NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

LEARNING ADVERSARY MODELING FROM GAMES

by

Paul Avellino

September 2007

Thesis Advisor: Craig H. Martell
Second Reader: Kevin M. Squire

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Learning Adversary Modeling from Games			**5. FUNDING NUMBERS**
6. AUTHOR(S) Paul Avellino			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Since ancient times, adversary modeling has been used during wargaming exercises in which military leaders have recreated past battles or simulated future battles in order to educate military professionals. Although the technology today is much different, adversary modeling still serves the same goals – to help military professionals learn tactics from past successes and mistakes. In the computer age, highly accurate models and simulations of the enemy can be created. However, including the effects of motivations, capabilities, and weaknesses of adversaries in current wars is still extremely difficult.

Limit Texas Hold'em poker, with many attributes similar to real-world warfare, is an excellent test-bed to study and improve adversary modeling. For example, stochastic outcomes which deal with multiple independent agents, deception, and acting amidst uncertainty, are some of the aspects of poker that closely resemble important aspects of warfare. These attributes make poker a better choice as a study platform than other traditional games, such as chess, where there is no deception or uncertainty.

The defined rules of poker provide researchers with a controlled environment to improve and test adversary-modeling techniques. Perfecting adversary modeling in poker will allow simulators to improve and generate more accurate models for wargames, giving warfighters the advantage in current and future battles.

14. SUBJECT TERMS Adversary Modeling, Opponent Modeling, Computer Poker, Artificial Intelligence			**15. NUMBER OF PAGES** 97
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

LEARNING ADVERSARY MODELING FROM GAMES

Paul D. Avellino
Captain, United State Marine
B.S. with Merit, United States Naval Academy, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Author: Paul D. Avellino

Approved by: Craig H. Martell, PhD
Thesis Advisor

Kevin M. Squire, PhD
Second Reader

Peter J. Denning, PhD
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Since ancient times, adversary modeling has been used during wargaming exercises in which military leaders have recreated past battles or simulated future battles in order to educate military professionals. Although the technology today is much different, adversary modeling still serves the same goals – to help military professionals learn tactics from past successes and mistakes. In the computer age, highly accurate models and simulations of the enemy can be created. However, including the effects of motivations, capabilities, and weaknesses of adversaries in current wars is still extremely difficult.

Limit Texas Hold'em poker, with many attributes similar to real-world warfare, is an excellent test-bed to study and improve adversary modeling. For example, stochastic outcomes which deal with multiple independent agents, deception, and acting amidst uncertainty, are some of the aspects of poker that closely resemble important aspects of warfare. These attributes make poker a better choice as a study platform than other traditional games, such as chess, where there is no deception or uncertainty.

The defined rules of poker provide researchers with a controlled environment to improve and test adversary-modeling techniques. Perfecting adversary modeling in poker will allow simulators to improve and generate more accurate models for wargames, giving warfighters the advantage in current and future battles.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

ACKNOWLEDGMENTS

Writing a thesis is no easy task. It takes the help of many people for many different reasons. I would like to thank the following people:

To my mother, Elizabeth Avellino, thank you for always encouraging me and supporting me and helping me to become everything I am today.

To Craig Martell, thanks for all the work and motivation you provided to help me complete my thesis.

To Kevin Squire, thanks for your insightfulness.

To Jane Lin, thanks for your encouragement to take my time in completing my thesis.

To Trent Bottin, our many discussion involving poker bots and techniques for adversary modeling were quite useful, as was the money I won from our poker sessions.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. HISTORY OF ADVERSARY MODELING

The importance of adversary modeling has been known for centuries. Sun Tzu [1], the 6th Century B.C. military strategist wrote:

> *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for each victory gained, you will also suffer a defeat.*

Adversary modeling has been used since ancient times in a military context during a process called wargaming. During a wargame, commanders seek to improve their battle plan by stepping through the plan with consideration given to the enemy's actions, reactions, strengths and weaknesses. Adversary modeling is conducted by an intelligence officer who has studied the enemy's capabilities and whose goal is to defeat the commander's plan so as to improve the plan.

Besides military applications, adversary modeling is used in a wide variety of areas. For example, in the computer-security realm, network-security professionals frequently create models of potential attackers in order to help them identify when their systems are being attacked. Additionally, adversary modeling has been studied and shown to improve bot performances in games such as Scrabble and RoShamBo [2],[3],[4].

1. Pre-Computer Adversary Modeling

Games like Go and Chess were used teach soldiers competence in battlefield situations. In these games, adversary modeling is not as important because they are perfect information games where all elements of the game (i.e., game board and game pieces) are known to all players. However, in actual wargaming situations, only limited information about the enemy is known and the rest must be inferred by an intelligence officer. Using the simplest adversary model, the intelligence officer acts as a friendly commander would act. While this approach does help find some weaknesses in a plan, it is far from being realistic. A much better model would simulate the enemy’s actions according to that enemy’s own doctrine. Although the benefits of this model are enormous because the enemy actions can reflect the leadership of a specific enemy commander, it necessitates a thorough understanding of the enemy commander’s tactics and observations obtained through vigorous analysis from many previous battles.

2. Computational Approaches

Since the advent of computers, wargaming has improved through more complex modeling and simulations. Using a computer and simulated battles, models of friendly and enemy units can fight with no loss of life, equipment, or other valuable resources. An accurate knowledge of an enemy’s doctrine, tactics, and motivations can tremendously improve the accuracy of these models and simulations. These modeling and simulation techniques have been incorporated into a commercial setting with the popularity of video games. Today, countless video games simulate old battles or create fictional or fantastic scenarios allowing players to wage battles with different tactics.

B. IMPORTANCE OF ADVERSARY MODELING

In all of the situations described above, highly accurate models of opponents increase the utility of the game. In commercial computer games, this makes a more realistic and higher selling game. In the wargaming scenario, a better model of the enemy helps create a better plan to defeat the enemy.

1. Military and Intelligence Community Adversary Modeling

During the Cold War, adversary models were simpler than they are today because Soviet doctrine was relatively well known. Battles and wars could be simulated during the wargame based on knowledge gleaned from past battles, known tactics and commanders, and obvious motivations and morale of the soldiers. Since the end of the Cold war and the beginning of the War on Terror, adversary models have become increasingly difficult to create accurately. Not only do motivations of a terrorist differ greatly from the motivations of a soldier fighting for his state, motivations of different terrorist groups can be vastly different from each other as well. For these reasons, modeling in this new age of warfare is very difficult.

2. Poker Adversary Modeling

The game of poker provides an excellent test-bed for adversary modeling. Poker is a game containing stochastic events, imperfect information, multiple competing agents, and deception. Like the real-world scenario of warfare, adversary modeling substantially improves performance in a poker game.

a. Introduction to Poker

In our studies, we use Limit Texas Hold'em Poker. The game is played with blind bets that players must make before cards are dealt. The first person to the left of the dealer begins with a bet called the "small blind." The person on their left follows the small blind with a bet called the "big blind," which is twice the size of the small blind. These bets, similar to an ante, are used to instigate action, or encourage others to bet. All subsequent bets and raises in the first to rounds are the size of the big blind.

A hand begins with each player being dealt two cards, called "hole cards," only known to that player. The blinds are considered legal bets; therefore, the person to the left of the big blind is the first person to act after looking at their hole cards. This person now has three options – fold, call, or raise. A "fold" means that the player does not wish to continue and opts out of the hand. A "call" means that the player wishes to play for the number of bets that has already been established (in this case one – the big blind). A "raise" means that the player wishes to increase the number of bets from one (the big blind) to two (twice the amount of the big blind). This concept of the number of bets is sometimes referred to as "bets-to-go" or "bets-to-call." Two bets-to-go simply means that all players who want to remain in the hand must pay two bets.

Play continues around the table until all players have either folded or called the highest raise. (Note: rules dictate that all betting rounds are capped at four bets.) If only one player remains, that player wins all the money in the pot and does not have to show their cards. The action up to this point is referred to as "pre-flop."

The "flop" is when three community cards (also called board cards) are placed face up in the center of the table. These cards are used by all players remaining in the hand. All remaining action is referred to as "post-flop." At this point, another round of betting begins. The first player remaining in the hand to the left of the dealer acts first. He can "check" or "bet." A check means that the player does not want to bet, and since no one else has bet, the player does not have to fold. A check keeps the game at zero bets-to-go while a bet makes it one bet-to-go. The betting continues as before, until everyone has folded or called the highest bet, or until only one player remains. Again the betting is capped at four bets-to-go. Now, a fourth community card, called the "turn," is dealt. This is followed by another betting round; however, all bets for this round and the final betting round are twice the size as the bets in the first two rounds. Finally, the "river" is the fifth and final community to be dealt. Following the river, there is a final betting round. At the end of this betting round, if more than one player remains, there is a "showdown" where the remaining players' cards are revealed. The highest five-card poker hand—five cards can be taken from any combination of the player's two hole cards and the five community cards—wins the pot. The hand is now over, and the dealer position is moved one seat to the left to initiate a new hand.

For simplicity, player's actions can be viewed as three choices: raise, call or fold. Bets and raises can be abstracted together and called a raise. A bet is simply a special case of a raise when the betting round is zero bets-to-go. Similarly, a check and call can be abstracted to a call, the check being a special case of a call when a player does not want to increase the number of bets-to-go from zero.

b. Importance of Adversary Modeling in Poker

Adversary modeling is a vital part of maximizing your play in poker. Research has shown that the game-theoretic optimal solution does not necessarily result in the best poker player [5]. Game theory approaches result in good but defensive play, where a player will never lose big, but they will also never win big. A good model of a poker adversary will allow us to exploit their weaknesses, thereby allowing us to win larger amounts of money.

C. MOTIVATION AND PURPOSE OF STUDY

Poker allows us to improve adversary-modeling techniques in a structured domain. Not only does poker sufficiently limit the domain with its rule set, its stochastic elements and hidden information provide a high resemblance to real-world adversarial situations, providing an accurate test-bed for adversary-modeling research.

In poker, every opponent has hidden information. More specifically, their hole cards are known only at the end of a hand, if at all. To apply this concept to warfare, it is evident that enemies have secrets. For example, the number of members in a terrorist cell is hidden and can change frequently, making that information impossible to know at all times. The dealing of cards is a stochastic event, which can be comparable to the numbers of disaffected youths that could be influenced by terrorist rhetoric. The strength of a player's hand can be determined and compared to the other possibilities of an opponents hand based on the community cards. Correspondingly, the strengths of terrorist groups might be calculated and compared. The number of bets-to-call could parallel the cost of military or political actions. In poker, "pot odds" is a measure of the reward of an action compared to the cost of that action and could be analogous to many military operations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. RELATED WORK

In the last decade, an increasing number of researchers began studying poker. For the last two years, a poker bot competition has been part of the annual Association for the Advancement of Artificial Intelligence (AAAI) convention. The fixed nature of this game (e.g. rules, betting actions) allows researches to build and improve adversary modeling techniques that can then be used in other domains. Adversary modeling is an important aspect of successful poker bots.

A. THE UNIVERSITY OF ALBERTA'S COMPUTER POKER RESEARCH GROUP

The University of Alberta's (U of A) Computer Poker Research Group (CPRG) conducted the seminal research in this field. In [6], Billings provides a concise synopsis of the major accomplishment of the CPRG. Perhaps most importantly, they established a publicly available corpus of poker game data that can aid in adversary-modeling experiments. They studied limit Texas Hold'em—recently focusing on heads-up games involving only two players.

Their research began with poker bots that are derived from a rule-based system. As is typical in artificial intelligence, this method has only limited effectiveness while the rules and knowledge base increase rapidly. The CPRG then attempted to calculate optimal play game theoretically. Finally, the CPRG experimented with using game-tree search methods to make decisions that result in the highest expected value. Varying degrees of adversary modeling are attempted by the CPRG, as discussed below.

1. Knowledge-Based Poker Player

The first iterations of the U of A's CPRG's poker bots used knowledge-based artificial intelligence to establish a baseline. Only average poker play was attainable before the knowledge base and rules became too large and complex. The adversary modeling performed in this poker bot was based on observed statistics. The crucial information to deduce is the adversary's hole cards. In the CPRG's studies, the opponent's hole cards are abstracted into 169 distinct hands. There are 13 different ranks, Two through Ace, and the cards are either suited or unsuited—making 169 distinct hands.

The simplest starting point for the probability of an adversary's hole cards is to assume a flat probability distribution function. This will provide a baseline, but will not correctly represent the probability of an adversary playing those hands because most players will play "better" hands with more probability than "worse" hands. The key variable is to determine which cards an opponent deems "better."

Using the "reasonable man" approach, the CPRG developed a generic adversary model (GOM) to infer which hole cards an average player is going to play. Billings et al. calculate an income rate, which is the expected value, for each possible pair of hole cards using simulations in [8]. Obviously, a "reasonable man" is less likely to play hands that result in a negative income. They assign probabilities to each of the 169 starting hands that are based on the calculated income rate of that hand. As the play of a hand unfolds, they adjust these probabilities based on actions in a hand. For example, if the adversary raises, the probabilities assigned to the hands with high income rates are increased, while the probabilities for the hands with low income rates are decreased. The increases are done based on rules that are applied to all players. However, not all players act as this GOM does. Some players are attracted to straights and flushes and are thus more likely to play cards that have a better chance of making those hands.

The CPRG performs specific opponent modeling (SOM) by changing the weights differently for each individual adversary. For example, if an adversary usually bets with a flush draw, their algorithm will increase the probabilities of those hands that give the adversary a flush draw. In order to deduce the probabilities to use at the start of a hand for a specific adversary, the CPRG maintains counts of betting frequencies in certain contexts of the game. As discussed in the introduction to poker, there are three actions: bet, call or fold. Their system tracks the frequencies of these actions in twelve different contexts: based on the betting round (pre-flop, flop, turn, river) and the number of bets-to-call (zero, one and two or more). Over time, these frequencies would begin to evolve and could lead one to make assumptions about an adversary. For example, if a player bet 35% of the time after the flop when there are zero bets-to-call, one could assume that the adversary would bet with the top 35% of hands, or the top 30% of hands and the other 5% based on strong drawing hands.

For pre-flop frequencies, these percentages are mapped back to the income rates. Post-flop, the frequencies are mapped to a hand strength based on possible adversary hole cards combined with the board cards. In [8], the CPRG admits that this method is flawed because it is based on the CPRG's calculations of income rates and hand strengths, which may be different from how the adversary calculates the strength of their hand.

In [9], the CPRG improved this method of adversary modeling based on the results of experiments with Artificial Neural Networks (ANNs). They used 19 different aspects of the game context as inputs to the ANN which would then produce a likelihood of a raise, call, or fold from an adversary. They determined that ANNs were good at filtering out noisy aspects of game contexts, but required too many historical hands before becoming accurate. Thus, ANNs are not feasible for the real-time nature of poker. However, they did ascertain that "last bets-to-call" and "last action" were important factors for an adversary's decision. These two dimensions of the game were added to the statistical model described above which produced improved results.

In the methods described above, there is minimal use of the board cards in the context of the game, which seems to be a conspicuous weakness.

2. Game Theoretic Methods

The CPRG devotes time to finding the game-theoretic optimal solution at each decision node. They apply a randomized mixed strategy to the adversary's actions. With no adversary modeling done in these experiments, the actions of the poker bot are only based only on known cards. The play of their bot improves significantly over the knowledge-based system and is even able to initially play well against a professional poker player. However, given more time, the professional is able to discover weaknesses and can exploit the bot [5].

3. Game Tree Search Methods

In their next set of experiments, the CPRG employs methods that search game trees in order to maximize the expected value (EV) of their decisions [10],[11]. In their game tree, there are four different types of nodes: chance nodes, adversary decision nodes, program decision nodes and leaf nodes. The chance nodes simply relate to the possible cards that could follow based on the known cards up to that point. The program decision nodes are where the program decides which action will result in the highest EV, with some variability added to disguise the program's play. The adversary decision nodes are an estimated probability that the adversary will take each action: raise, call, or fold. This probability is based on counts of past actions at the corresponding point in the game tree and is in no way affected by the cards the adversary holds or the community cards, even if the previous counts ended in a showdown, where the adversary's cards are revealed. The leaf nodes contain the EV of that node and the probability of winning the pot. The probability of winning the pot is determined using a histogram of previous hand strengths that the adversary has shown at showdowns that correspond to that leaf in the game tree. The program will compare its hand strength at that leaf to the hand strength histogram of the adversary to determine the probability of winning the hand.

This method uses abstractions when the game tree is incomplete in order to be effective when little information is known. One abstraction is obtained by using all branches of the game tree that have the same number of bets and raises, ignoring when the bets and raises are made. Another, finer-grained version of that abstraction uses all branches with the same ordered pair of the total bets and raises of both players. A more coarse-grained abstraction is simply the total number of bets and raises by both players. Another form of abstraction considers only the final size of the pot. In their experiments, the CPRG uses a combination of all of these abstractions. The abstractions are weighted stronger for the finer granularity of the abstraction and a mixture of all is used based on the weighting system. Generic adversary models are used as defaults until enough hands are recorded to make the specific adversary modeling precise.

This method completely ignores the fact that the board cards will factor into the adversary's decision making process. Additionally, a high computation time is needed for all decisions because the entire game tree must be searched to completion for each decision.

4. Bayes' Bluff

In [12], Southey, et al, experiment with a probabilistic model for opponent modeling. Each player has a strategy that is known only by them. Each player also has an information set for each hand consisting of the cards visible to them. Using Bayes' Rule, the probabilities of an opponent playing different strategies are calculated using the observations of all hands—hands that go to a showdown and hands that are folded. Next, the authors use the posterior distribution over the strategies to determine the best response to an opponent in the current hand. The best response is the action that results in the highest expected value. The authors tested this method against various other poker bots. The results show that this model is effective in countering an opponent's strategy in as little as 200 hands.

B. OTHER RESEARCH

As poker increases in popularity revealing more complexities, other researchers have joined in with experiments of their own. The most influential methods for the research described in this thesis follow.

1. Carnegie-Mellon University Method

In [13],[14],[15], Gilpin and Sandholm describe a method of calculating the game theory equilibrium and then use Bayes rule for predicting the hole cards of an opponent. Offline, they compute optimal strategies for playing the pre-flop and flop rounds. They first use automated abstraction techniques to condense the complexities of the game. Then, they perform equilibrium computations using linear programming to calculate the expected value of future stochastic events (cards dealt in the upcoming turn and river rounds) without regards to future bets. During the turn and river rounds, the authors apply Bayes' rule to calculate the probability of all possible hole cards based on the computed strategies and the observed actions in the prior rounds. This method is computationally expensive but accounts for game context more than many other methods described in this thesis. However, the authors do not use any information from previous hands to influence action of the bot. Although their poker bot did win small amounts of money in their early experiments, the authors could not show that their poker player preformed better than the expected variance of Texas Hold’em [13]. Later results in [14],[15] show that their improvements produced a statistically significant win rate.

2. Bayesian Networks

There have been several researchers who conducted experiments using Bayesian networks in [16],[17],[18],[19]. Although Korb, et al, and Boulton [17],[18] describe research conducted using another form of poker (Five Card Stud), it is useful to discuss their use of Bayesian networks which is the basis for later models that Carlton describes in [19].

In [20], Russell and Novrig describe a Bayesian network as a directed acyclical graph in which each node represents a random variable and each arc represents influence of one node on another node. Conditional probability tables are used to quantify the effect that parent nodes have on the child. The biggest drawback of using Bayesian networks for modeling opponents is the need of these defined dependencies. The authors of [16] use dependencies among such game attributes as position, action, pot odd, hand strength, etc. However, not every poker player uses the same variables nor is everybody’s dependencies the same as the authors’. This is evidenced by fact that the Bayesian networks shown in [17],[18],[19] use different nodes and arcs in their models.

In [19], Carlton creates a generic opponent model by using self-play to initialize the conditional probability tables. This bootstraps the Bayesian network in order to be more effective at the start of play against an unknown opponent. Then, a generic opponent model is created by editing the conditional probability tables according to the actions of a specific opponent during game play.

The authors of these papers show little accuracy in their results. Carlton showed the best results in [19], but was still not able to beat human opponents or the state-of-the-art poker bots. These authors suggest that a more complex Bayesian network or a dynamic Bayesian network may yield better results. Dynamic Bayesian networks allow the relationships between the nodes to change at different stages of the game, but the dependencies still need to be defined.

C. RESEARCH CONDUCTED IN THIS THESIS

1. The Use of Game Context

Most of the methods described above made little use of the context of the game. In poker, this would be the community cards and the actions taken given these community cards. Additionally, the cards revealed at showdown can be rolled back to give insight into the decision made earlier in the hand.

The methods that do use game context use Bayesian Networks where the variables and dependencies are hard-coded. This, as discussed above, does not work well against opponents who do not use the same variables and dependencies.

2. Hidden Markov Models

Hidden Markov Models (HMMs) have an advantage over the methods describe above. Using HMMs, one can take into account the entire context of the game without defining the variables and dependencies that an opponent might use to make decisions. The hidden states in the HMM can represent the variables and dependencies used by an opponent to make his decisions. Furthermore, training the HMM for different opponents over different sequences of actions during the hands of a game allow the HMM to accurately represent different opponents.

III. DATA GATHERING AND DESIGN OF EXPERIMENTS

A. DATA GATHERING

1. University of Alberta's Corpus

The University of Alberta collected data from IRC-based poker rooms for years. This data is available online [21]. This corpus is used for much of the research conducted by the University of Alberta and other scientists. The corpus consists of a separate folder for each month of play. Within each month folder there is a hand database file, a hand roster file, and a player database folder.

The hand database file lists, from left to right, a timestamp for the hand, the position of the dealer, the hand number, the number of players dealt in the hand, the number of players, the amount of money in the pot at the flop, turn, river, and showdown, and the community cards that were dealt (See Figure 1).

797211363	1	197	7	3/95	3/125	3/185	1/205	2c 9s 7c 9d Js
797211458	1	198	9	2/45	0/0	0/0	1/55	Jd Qs 8s
797211529	1	199	9	3/65	2/85	0/0	1/105	Qh 2c 5c 6s
797211616	1	200	8	4/80	3/140	3/200	2/280	Ts 4d 4c 7s Jc
797211721	1	201	8	4/80	4/120	2/160	2/200	7d 6h 5d 9c Tc

Figure 1. Example hand database information.

The hand roster, shown in Figure 2, consists of the timestamp for each hand, the number of players dealt in that hand and the user name of each player dealt in that hand.

```
797210868  9 Quick Winner777 derek greg gunner jims johnr sagerbot shinner
797210948  8 Winner777 derek greg gunner jims johnr sagerbot shinner
797211062  8 Winner777 deadhead derek greg gunner jims sagerbot shinner
797211160  7 deadhead derek greg gunner jims sagerbot shinner
797211251  7 deadhead derek greg gunner jims sagerbot shinner
797211363  7 deadhead derek greg jims k^man sagerbot shinner
```

Figure 2. Example hand roster information.

The player database folder contains a separate file for each player who played at least one hand during that month. These files list the following information for each hand in which the player participated (See Figure 3): their name, the timestamp of the hand, the number of players dealt in that hand, their position relative to the "dealer" position, their actions, the amount of money they had at the beginning of the hand, the amount they contributed to the pot, the amount they won from the pot, if any, and their hole cards, if they were involved in a showdown.

sagerbot	797210868	9	1	Bk	b	b	kc	1740	60	0	Qd Jc
sagerbot	797210948	8	1	Bc	kf	-	-	1680	20	0	
sagerbot	797211251	7	7	c	r	b	b	1660	80	205	9s 8s
sagerbot	797211721	8	2	Bc	c	b	b	1785	70	200	8h Ah
sagerbot	797211886	8	8	c	c	f	-	1910	30	0	
sagerbot	797212372	6	2	Br	b	b	-	1880	60	140	
sagerbot	797213334	10	2	Bk	b	-	-	1890	20	55	
sagerbot	797213396	10	1	Bc	br	br	b	1925	130	330	Jd 7d

Figure 3. Example player database information.

All information needed for this research was ascertained using the above files.

In addition to the corpus of data, the University of Alberta provides basic, poker related code [22]. They have java source code files for a card, a deck, a hand, and a hand evaluator. The first three are simple classes to represent important concepts in the game. The hand evaluator assigns an integer to every possible five-card hand such that a higher hand will be assigned a larger integer and two equal hands will be assigned the same integer. This class returns the integer representing the strength of the hand for any input of cards numbering between three and seven.

2. Creating Hand Histories from Corpus

Perl code was used to create hand histories for players with the most hands, which is based on the size of the player's file in the player database. Chosen at random, data from May, 1995 was used in these experiments. The hand histories are files that contain all the information about the actions of all the players in each hand in which the target player participated. This data was mined from all the other player database files in the given month.

3. Composition of the Action Vector

For this research, an action vector was created for each action performed by the target player (See Figure 4). The action (ACT) was limited to raise, call, or fold, based on arguments described in the explanation of poker in Chapter I. The following information about the board cards was used: board score (BS), probability of a straight draw (PSD), the probability of a flush draw (PFD), the probability of a straight (PS), the probability of a flush (PF), and the Boolean concerning if the board contains a face card (FC). This data is set at zero for all actions that occur pre-flop. The board score is an integer returned from the University of Alberta's hand evaluator class that represents the strength of the board cards alone.

When a poker player has a potential to make a good hand but needs another card, the player is said to be on a "draw," (e.g. four cards of the same suit is called a flush draw). Flushes, straights, and draws to straights and flushes were modeled using probabilities. To obtain a probability of having a flush or a straight, every possible two-card combination of the remaining cards that when added to the current board cards makes a straight or a flush is divided by the number of all possible two card combinations to obtain a probability. A similar method is used to determine the probability of a draw, except a third card is added to represent the next board card to be dealt.

In addition to the board information, the following information is tracked for every action: the number of players still in the hand who act before the target player (PA), the number of people who act after the target player (PB), the number of bets-to-call (BTC), the pot odds (PO), and the amount of money the player has when he performs each action (POT). "Pot odds" is a term that represents a player's reward-to-risk ratio and is the quotient of the amount of money already in the pot and the amount to call the current bet.

The final information in the action vector is only available when the target player reveals their cards at a showdown. These showdown cards are used for all actions that the player conducted in that hand to determine the strength of the players hand relative to all possibilities (HS). For pre-flop strength, a lookup table was used that contains probabilities of having the best two-card hand. This probability is based on research by Sklansky [23], a professional poker player, and Billings [6]. After the flop, the hand evaluator class discussed above is used along with the method similar to the one used to determine the possibility of a straight or flush. Every possible two-card combination is added to the board cards. The number of combinations that return a higher integer than the player's hand is divided by the total possible combinations to obtain a number between one and zero. This number is used to represent the strength of the player's hand.

BS	PSD	PFD	PS	PF	PA	PB	BTC	PO	HS	POT	FC	ACT
0	0.0	0.0	0.0	0.0	1	0	1	0.0	0.650	1740	F	B
1995	0.0683	0.0895	0.0	0.0	0	1	0	0.0	0.8501	1740	T	B
25980	0.7487	0.0953	0.2836	0.0	0	1	0	0.0	0.9391	1730	F	C
337772	0.7782	0.1017	0.2960	0.0	0	1	0	0.0	0.9391	1720	T	F
337772	0.7782	0.1017	0.2960	0.0	1	0	1	0.1666	0.9391	1720	F	R
0	0.0	0.0	0.0	0.0	2	1	2	0.1538	0.0	1680	F	C
2135	0.0683	0.0895	0.0	0.0	0	2	0	0.0	0.0	1660	T	F
2135	0.0683	0.0895	0.0	0.0	2	0	1	0.1	0.0	1660	T	F
0	0.0	0.0	0.0	0.0	1	2	2	0.3076	0.0	1660	F	B
0	0.0	0.0	0.0	0.0	1	1	1	0.2857	0.0	1660	F	C

Figure 4. Example action vectors

4. Data Mining Hand Histories for Information

Java code was written to step through the hand histories to make the action vectors described above. All the vectors for a given hand are stored in one file. These files are labeled with a number and the strength of the hand at the river. The strength of hand is defined as high, medium, low, and folds. Folds are hands that were folded and the hole cards remain unknown. For the remaining categories, the hand strength, as described in the previous section, is used. High is defined as 0.70 and higher. Medium is defined as greater than or equal to 0.40, but less than 0.70. Any hand lower than 0.40 is defined as low. An additional file containing every vector is created and is used to determine clusters of hands for use in the following experiments.

B. DESIGN OF EXPERIMENTS

1. Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model used to describe the state of a changing environment [20]. The states represent different values of discrete random variables over time. If one assumes a Markov process, a process in which the current state only depends on the previous state and not earlier states[1], an HMM is useful when there is noise or uncertainty in the environment. In an HMM, the states are hidden or unknown but determine the observable evidence emitted by the model.

a. Structure of the HMM

An HMM consists of a set of states, a start distribution, a transition matrix, and an observation matrix. The states are used to represent the hidden (or unknown) variables in a random process. The start distribution shows the probability of beginning in each state. The transition matrix contains the probability of moving from one state to any other state in the model. An HMM may allow only one path through the model, a linear model with no jump-ahead, or it may be possible to go from any state to any other state, an ergodic model, or some

[1] This describes a first order Markov process, in a second order Markov process, the current state only depends on the previous two states, and likewise for third and fourth order processes.

variation in between these two models. The observation matrix describes the probability of seeing a given observation in a particular state.

There are three tasks normally associated with an HMM:

- Evaluation: given the parameters of the model, compute the probability of a given observed sequence using the forward-backward algorithm.
- Decoding: given the parameters of the model, compute the sequence of states that most likely generated the observed sequence using the Viterbi algorithm.
- Learning: given an observed sequence or set of sequences, calculate the model that best explains the observation sequences using the Baum-Welch algorithm.

b. Training and Testing

For the purposes of the experiments in this thesis, it is not necessary to compute the sequence of states that generate the observations. In abstract terms, the states of the HMM are supposed to model what the player believes about the strength of his hand. The observations are his actions (raise, call or fold) and the game context at the time of his actions. The Baum-Welch algorithm is used to train the HMMs used in these experiments. Once the HMMs are trained, the forward-backward algorithm is used to determine which HMM was mostly likely to produce a given sequence.

2. Using Hidden Markov Models

Experiments with HMMs were conducted in Matlab. For k-means clustering, fast k-means code for Matlab was used [24]. HMM Toolbox for Matlab is used for all of the HMM operations [25].

a. Vector Quantization of Game Context

K-means is an algorithm for grouping large amounts of data into *k* different groups. The objective is to minimize the total distance from every data point to one of the centroids. To accomplish this task, *k* centroids are chosen throughout the space at random. Then, each data point is assigned to the closest centroid, creating *k* clusters of data. Next, ignoring the current centroids, centroids for the *k* groups are re-calculated and placed at the center of each of the *k* clusters. Again, each data point is assigned to the closest centroid. The algorithm repeats a given number of times or until the distance between successive centroids is below some minimum threshold. Each of the *k* centroids is labeled with an integer, 1 through *k*. The algorithm returns the integer, *k* representing the centroid closest to each of the data points.

For these experiments, k-means was used to reduce the number of different sequences used to train the HMMs. This is similar to assuming that hands would be played similarly during similar situation in a poker game. The following numbers of centroids were used in the experiments in this thesis: 50, 75, 100, 175, 250, and 500. Two dimensions of the action vector are eliminated before the clustering process: 1) the Boolean variable for face card present (FC), and 2) the action (ACT) - raise, call, or fold. The k-means algorithm returns the 11 dimension cluster centroids and an integer (1 through *k*) representing that centroid. For simplicity, the integer representing the centroid is used in the experiments instead of the vector. In order to retain the information for FC and ACT that was not used in clustering, digits are appended to the end of the integer representing the cluster center. First, one digit is appended to represent FC – a "0" for false and a "1" for true. Finally, the second digit appended represents the action - the label "0" means fold, "1" stands for call, and "2" represents raise. At this point, each action vector is represented by one integer. For example, the experiments with 50 centroids uses integers ranging from 100 to 5013; for experiments with 250 centroids, these integers range from 100 to 25013.

b. Representing a Hand for Training and Testing HMMs

In order to train the HMM, the input training sequences must contain all the actions of one hand on a single line. Furthermore, each hand must be of equal length; therefore, each hand is padded with integers to ensure that each sequence is of equal length. Since zero cannot be used as an input, an integer higher than any possible value of an action vector is used – 5014 for the 50-centriod experiment and 25014 for the 250-centroid experiments are examples. Any hand in which the player's first action was a fold was not used for training or testing. Figure 5 shows ten example hands from the 100-

centroid HMM. Notice that all hands end with several instances of padded integer – 10014 in this case. In the first hand in Figure 5, the first action vector is represented by 2601. 26 is the label of the vector quantized game context, the value of the Boolean FC is 0 and the action (ACT) is a call, represented by a 1. The second action of the hand is represented by the 2612: 26 for the game context, 1 for the presence of a face card, and 2 for the action of a raise.

2601	2612	8412	4611	4611	10014	10014	10014	10014	10014	10014	10014
2601	8202	8202	8202	8211	8211	10014	10014	10014	10014	10014	10014
2602	2601	8202	8202	8202	10014	10014	10014	10014	10014	10014	10014
9602	9601	9601	3701	8711	10014	10014	10014	10014	10014	10014	10014
9601	9612	9611	4111	4111	8912	10014	10014	10014	10014	10014	10014
9602	9601	3701	5901	10014	10014	10014	10014	10014	10014	10014	10014
2601	2612	2611	8111	8111	5211	10014	10014	10014	10014	10014	10014
2601	2611	8411	8811	10014	10014	10014	10014	10014	10014	10014	10014
4801	4812	9312	9912	10014	10014	10014	10014	10014	10014	10014	10014
2602	2601	3812	3811	412	10014	10014	10014	10014	10014	10014	10014

Figure 5. Example training and testing data.

c. Experiments with Four HMMs

The first experiment is to determine if HMMs are capable of categorizing a hand as a high, medium, low, or fold hand. To accomplish this, eight files are created for the player, two for each category of hands: high, medium, low, and fold hands. Eighty percent of the hands are placed in training files and twenty percent are placed in testing files. The HMMs used during these experiments have either four or eight states. The models used were ergodic; transitions are allowed from every state to any other state. Four HMMs were trained, one corresponding to each category of hand (high, med, low, and fold) using the files containing eighty percent of the hands. The held-out twenty percent are then used to test this process. For observation sequences, the first action of a hand is used, then the first two actions are used, and so on, until the entire hand is used for a sequence. At each point, the forward-backward algorithm was used for each of the four HMMs in order to determine which HMM was mostly likely to produce the sequence so far.

d. Experiments with Three HMMs

A second set of experiments was conducted similarly to the method above. The only difference was that no fold data was used. Therefore, only three HMMs were trained. The HMMs were used to attempt to determine a high, medium, or low hand.

e. Experiments with Two HMMs

In the third set of experiments, a different method was used. Instead of one HMM per category, only two HMMs were used for each experiment. These experiments attempt to classify hands as fold or not-fold, high or not-high, medium or not-medium, and low or not-low. As an example, in the fold or not-fold experiment, all of the high, medium, and low data was put into one file and used to train one HMM instead of three different HMMs, *mutatis mutandis* for high, medium, and low experiments. Again, the data was separated into eighty percent training data and twenty percent held-out testing data. Again, the forward-backward algorithm is used on each sequence of the testing data to determine which of the two HMMs most likely produced the sequence.

IV. RESULTS AND ANALYSIS

A. RESULTS AND ANALYSIS

Accuracy, precision, recall, F-score and baseline F-score were all used to evaluate the performance of the HMMs. Accuracy is the number of predictions correct divided by the total number of predictions. Precision is the proportion of the predictions of X that were correctly labeled—X being the possible categories of high, medium, low, or fold hands. Recall measures the proportion of X's in the corpus that were correctly labeled X. The F-score is the harmonic mean of recall and precision given by the following formula, where F is the F-score, P is the precision, and R is the recall:

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

The F-score is used to balance the recall and precision. In order to attain a high F-score, both the recall and precision must be high; therefore, one cannot improve one measure at the expense of the other measure. The baseline F-score is calculated using the F-score formula as if every prediction was X. Therefore, the recall will always equal one and the precision will be proportional to the frequency of X. This is used too measure whether or not the performance of the HMM is better than chance. The baseline F-score is referred to as baseline for the remainder of this thesis. If the F-score is higher than the baseline, the HMM can predict better than chance and assuredly the data contains information that can be used for prediction.

The highest accuracy of the HMMs in this thesis was around 85%; however, most HMMs only attained 60% accuracy. Although the accuracy is not consistently high, many scores were significantly above the baseline score. Additionally, a high precision when predicting fold hands and high hands – especially in hands with many actions - was achieved in the experiments. The following paragraphs provide highlights of the results, with the full results given in Appendix A.

1. Experiments with Four HMMs

The HMM with eight states that used 100 centroids performed the best. The tables in Section 1 display the results of this HMM. As stated in the experimental design, the HMM made a prediction based on the first action in a hand, then the first two actions in a hand, then the first three actions in a hand, and so on, until the end of the hand. The results for all predictions are given in Table 1. Although the accuracy is around 50%, the scores are significantly above baseline for all categories except folds.

All Actions - 1880 Predictions				
Accuracy:	0.5122			
	Fold	Low	Med	High
Recall	0.5641	0.2105	0.3368	0.5392
Precision	0.7657	0.1491	0.1625	0.5207
F-score	0.6496	0.1746	0.2192	0.5298
Baseline	0.7110	0.1143	0.1862	0.4437
±Baseline	-9%	+53%	+18%	+19%

Table 1. Results for 8-state, 100-centroid four HMM experiment for all predictions.

It should be expected that with more information available, the HMM would perform better. In order to test this hypothesis, the performance at certain points in each hand is analyzed. The prediction based on the first action in a hand can be expected to be low, as there is very little information. However, the accuracy of the first prediction is 55% (See Table 2), which is better than the overall accuracy. The HMM never makes a "low" prediction based on the first action. This is not out of the ordinary, as low hands can easily be confused with fold hands. In fact, of the 27 low hands, 24 were predicted as fold hands based only on the first action.

First Action - 500 Predictions				
Accuracy:	0.5520			
	Fold	Low	Med	High
Recall	0.7231	0.0000	0.2500	0.2885
Precision	0.7015	0.0000	0.1358	0.3571
F-score	0.7121	0.0000	0.1760	0.3192
Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-10%	-100%	+9%	-7%

Table 2. Results for 8-state, 100-centroid four HMM experiment for the first prediction in each hand.

As play continues in a hand, a player will have more actions to use in order to judge the strength of an opponent's hand. We hypothesized that using the first three actions of a hand to make a prediction should improve the performance of the HMM. However, Table 3 shows that the accuracy drops considerably. The performance on fold hands is extremely low and many medium hands are mistakenly labeled as high hands. Note that if the opponent does not perform three actions in the hand, the hand is not included in this table. The third action of a hand is likely to be just after the flop where the strength of a hand changes considerably. This may explain why the performance drops at this point in the hand.

Third Action - 284 Predictions				
Accuracy:	0.3908			
	Fold	Low	Med	High
Recall	0.1455	0.4074	0.3721	0.6539
Precision	0.5333	0.2200	0.1861	0.5763
F-score	0.2286	0.2857	0.2481	0.6126
Baseline	0.5584	0.1736	0.2630	0.5361
±Baseline	-59%	+65%	-6%	+14%

Table 3. Results for 8-state, 100-centroid four HMM experiment for the first three actions.

The sixth action will typically be well after the flop when hand strength is relatively stable. Accordingly, the performance of the HMM increases significantly over the performance based on the first three actions, (See Table 4). Again, if the hand does not contain six actions, the performance of the hand is not included in this table. Note that the precision of folds is approaching 90% while the precision of high hand is almost 85% at this point. This tells a player that if the HMM predicts a fold, it is 90% sure the opponent will fold, and if the HMM predicts high, it is 85% sure the opponent has a high hand. Being able to distinguish between high and fold at this stage in the hand is very important because there is likely to a large pot at stake. Making this distinction can earn a good deal of money or prevent the loss of more money. Furthermore, all of the medium hands that are mislabeled are called high hands and most of the mislabeled high hands are called medium hands. This indicates the predictions are close and perhaps changing the threshold between medium and high hands may improve the performance significantly.

6th Action - 53 Predictions				
Accuracy	0.6038			
	Fold	Low	Med	High
Recall	0.6667	0.0000	0.4000	0.6471
Precision	0.8889	0.0000	0.1333	0.8462
F-score	0.7619	0.0000	0.2000	0.7333
Baseline	0.3692	0.0727	0.1724	0.7816
±Baseline	+106%	-100%	+16%	-6%

Table 4. Results for 8-state, 100-centroid four HMM experiment for the first six actions.

Although there are only six hands that contain eight or more actions, Table 5 shows that a high precision is attainable in the fold and high categories.

8th Action - 6 Predictions				
Accuracy	0.3333			
	Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.2000
Precision	1.0000	0.0000	0.0000	1.0000
F-score	1.0000	0.0000	0.0000	0.3333
Baseline	0.2857	2.0000	2.0000	0.9091
±Baseline	+250%	-100%	-100%	-63%

Table 5. Results for the 8-state, 100-centroid four HMM experiment for the first eight actions.

Table 6 shows the results of only the last prediction of each hand. The last prediction of each hand uses all the actions in that hand -- be it two actions or eight actions -- to make a prediction. This table shows the highest accuracy for this HMM and a very high precision on fold hands. This is somewhat misleading because the fold action is part of the action vector and is always the last action in a fold hand. The fact that the F-score is not higher shows that the actions preceding the fold mathematically outweigh the fold action in many of the hands.

Last Action - 500 Predictions				
Accuracy:	0.6400			
	Fold	Low	Med	High
Recall	0.7108	0.3333	0.3636	0.6154
Precision	0.9352	0.3913	0.2000	0.4267
F-score	0.8077	0.3600	0.2581	0.5039
Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+3%	+251%	+60%	+46%

Table 6. Results for 8-state, 100-centroid four HMM experiment for the last prediction.

For these experiments, accuracy between 55% and 60% is common, with the accuracy generally increasing as the number of actions in the hand increases. Additionally, as the number of actions increases, the precision of the fold hands and high hands increases.

2. Experiments with Three HMMs

The HMM with four states and 50 centroids performed reasonable well and was consistently between 51% and 55% on accuracy. However, the results for the HMM with eight states and 100 centroids preformed the better in key areas described below.

Similar to the previous experiments, this method achieved an accuracy of 53% on all predictions. Low performs 19% better than the baseline score. Most of the mistakes in the high and medium categories are in the opposite category, again showing that a change in the threshold between these two categories may cause significant improvements. These results are shown in Table 7.

All Actions - 843 Predictions			
Accuracy:	0.5255		
	Low	Med	High
Recall	0.2807	0.3834	0.6287
Precision	0.2883	0.2731	0.7310
F-score	0.2844	0.3190	0.6760
Baseline	0.2382	0.3726	0.7774
±Baseline	+19%	-14%	-13%

Table 7. Results of 8-state, 100-centroid three HMM experiment for all predictions.

This time, as should be expected, the prediction based on only the first action is worse than the overall accuracy, (See Table 8. Similarly to the first action in the four HMM experiment, this model does not predict a low hand based on the first action.

First Action - 175 Predictions			
Accuracy:	0.4743		
	Low	Med	High
Recall	0.0000	0.3636	0.6442
Precision	0.0000	0.2388	0.6204
F-score	0.0000	0.2883	0.6321
Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-28%	-15%

Table 8. Results of 8-state, 100-centroid three HMM experiment for the first prediction.

The performance based on the first three actions is considerably higher—exceeding 58% (see Table 9). Furthermore, the recall and precision scores are higher in all categories here than those recorded in the four HMM experiment.

Third Action - 175 Predictions			
Accuracy:	0.5805		
	Low	Med	High
Recall	0.4444	0.3954	0.6923
Precision	0.5217	0.3269	0.7273
F-score	0.4800	0.3579	0.7094
Baseline	0.2687	0.3963	0.7482
±Baseline	+79%	-10%	-5%

Table 9. Results of 8-state, 100-centroid three HMM experiment for the third prediction.

The performance of the last prediction is right at the average for the three HMM experiments and performed much worse than the four HMM experiments (See Table 10). This is likely due to the fold data that is inherent in the last action of a fold hand, as discussed in the previous section.

Last Action - 175 Predictions			
Accuracy:	0.5086		
	Low	Med	High
Recall	0.3333	0.3636	0.6154
Precision	0.2647	0.2909	0.7442
F-score	0.2951	0.3232	0.6737
Baseline	0.2673	0.4018	0.7455
±Baseline	+10%	-20%	-10%

Table 10. Results of 8-state, 100-centroid three HMM experiment for the last prediction.

Except for predictions based on the first three actions, this method did not perform better than the four HMM experiment.

3. Experiments with Two HMMs

Accuracy is much improved in these experiments – exceeding 85% in some cases. This shows that given broader categories, we can improve our performance.

Similar to the above experiments, 100 centroids result in the highest accuracy. The accuracy for fold hands is about 67% based on all actions (See Table 11).

All Actions - Fold or Not - 1880 Predictions		
Accuracy	0.6718	
	Negative	Positive
Recall	0.6868	0.6596
Precision	0.6212	0.7215
F-score	0.6524	0.6892
Baseline	0.6192	0.7110
±Baseline	+5%	-3%

Table 11. Results for the 100-centroid fold or not-fold HMM for predictions based on all actions.

Low hands scored the lowest accuracy on the predictions based on the first actions and the highest accuracy in the last predictions. Table 12 shows that the first action is only able to discriminate low or not-low at a 39% rate. As expected, this is difficult to determine base solely on the first action of a hand.

First Action - Low or Not - 500 Predictions		
Accuracy	0.3900	
	Negative	Positive
Recall	0.3679	0.7778
Precision	0.9667	0.0656
F-score	0.5329	0.1210
Baseline	0.9723	0.1025
±Baseline	-45%	+18%

Table 12. Results for the 100-centroid HMM predictions for Low or Not-Low based on the first action.

Table 13 shows that as the hand progresses, it becomes easier distinguish low from not-low. In fact, this is where the highest accuracy is attained—exceeding 84%.

Last Action - Low or Not - 500 Predictions		
Accuracy	0.8460	
	Negative	Positive
Recall	0.8710	0.4074
Precision	0.9626	0.1528
F-score	0.9145	0.2222
Baseline	0.9723	0.1025
±Baseline	-6%	+117%

Table 13. Results for 100-centroid HMM for predictions of Low or Not-Low based on the Last Action.

Interestingly, Tables 14 and 15 show that medium and high hands are relatively easy to discriminate on the first action. For medium or not-medium hands, accuracy over 70% was attained.

First Action - Medium or Not - 500 Predictions		
Accuracy	0.7040	
	Negative	Positive
Recall	0.7259	0.4773
Precision	0.9350	0.1438
F-score	0.8173	0.2211
Baseline	0.9540	0.1618
±Baseline	-14%	+37

Table 14. Results for the 100-centroid HMM for predictions of Medium or Not-Medium based on the First Action.

When discriminating between high and not high, accuracy over 66% was attainable on the first action.

First Action - High or Not - 500 Predictions		
Accuracy	0.6620	
	Negative	Positive
Recall	0.7096	0.4808
Precision	0.8388	0.3030
F-score	0.7688	0.3718
Baseline	0.8839	0.3444
±Baseline	-13%	+8%

Table 15. Results for the 100-centroid HMM for predictions of High or Not-High based on the First Action.

Table 16 shows the best accuracy in all of the experiments described in this thesis. As with the 100-centroid HMM, the 250-centroid HMM performed best when determining low or not-low based on the last action of the hand. The accuracy here was over 85%.

Last Action - Low or Not - 500 Predictions		
Accuracy	0.8560	
	Negative	Positive
Recall	0.8774	0.4815
Precision	0.9674	0.1831
F-score	0.9202	0.2653
Baseline	0.9723	0.1025
±Baseline	-5%	+159%

Table 16. 250-centroid HMM for Low or Not Low predictions based on the last action.

B. SUMMARY

In general, our experiments were successful in the following areas. Precision increased significantly as increasing numbers of actions are made in a hand, specifically in fold and high hands. Most high hands that were mislabeled were called medium, and vice versa. This indicates that adjusting the threshold between these hand categories will improve performance.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

A new method for adversary modeling was explored in this thesis. There have been numerous experiments conducted on adversary modeling in a wide array of domains—to include poker—but none have used Hidden Markov Models in the manner described here. This thesis uses Hidden Markov Models to predict what an opponent thinks about the strength of his hand. First, data was collected from an online corpus and mined for the information about the hands of several individual players. Next, we choose 13 dimensions of the game of poker of which an opponent could use to judge the strength of his hand. These game contexts were clustered together using the k-means algorithm and then used to train Hidden Markov Models. Several models were used to determine the most likely model to produce a given sequence of a hand, i.e., predict the strength of the hand. Finally, precision, recall, and F-scores were used to evaluate the performance of the models. The methods in this thesis did not produce accuracy above 85% and was usually lower than 60%; however, most results were above the baseline, which means the predictions were better than random. Furthermore, late in hands the HMMs were able to make clear distinctions between fold hands and high hands—a distinction that will earn a large amount of money in the long run.

B. FUTURE WORK

1. Adjusting Hand Strength Thresholds for Hand Categories.

In addition to the work described above, other experiments were conducted using different thresholds for high, medium, and low hands. Additionally, more hands were used in the experiments, resulting in more hands with up to eight actions. In one set of experiments, the threshold for high hands was raised to 0.90 and the threshold for medium hands was raised to 0.70. In another set of experiments, the threshold for high was set to 0.85 and the threshold for medium hands was set to 0.65. In these experiments, there were at least 26 hands of at least eight actions; as opposed to the six hands with at least eight actions described in Chapter IV. Additionally, the distributions of hands in the high, medium, and low categories were evenly distributed in these new experiments. The predictions based on the first eight actions produced many high scores. All predictions were well above baseline. For fold hands, the F-score was 94%, with a recall of 100% and a precision of 89%. The precision from high hands was also 100% and the overall accuracy score was 69%. This indicates that adjusting the thresholds further could result in even better performances. Unfortunately, different thresholds might produce different results for each opponent - negating one of the greatest benefits of using HMMs.

2. Modeling Advanced Play in Poker

Misinformation is inherent in the game of poker. Many advanced players will "slow-play" some hands – the technique of playing a very strong hand weakly in order to extract more money from your opponent. The opposite of slow-playing is bluffing – playing a weak hand as if it were very strong in hopes of making your opponent fold. Another advanced technique is drawing to a strong hand – where a player who does not currently have a strong hand but can call or raise because of a high likelihood of getting a strong hand with future board cards.

Modeling these types of hands is extremely difficult. Some of the bluff and draw hands could end up in the fold category – if the opponent re-raises and then the bluffer fold, or if the drawing hand does not catch the draw and folds. Despite the difficulties, some data mining techniques could be used to classify hands into these categories. Then, these hands could be used to train and test more HMMs. Future experiments would involve high, medium, low, bluff, slow-play, draw, and fold hand categories with a corresponding HMM for each category.

3. Principle Components Analysis

In these experiments, the integer labels for the centroids were used instead of the centroids themselves. If the data point of the centroid contains valuable information, using the point instead of label for the point may improve the performance. Principle Components Analysis (PCA) is a technique used to analyze multidimensional data. PCA uses linear combinations of the original dimensions to convert the data into a coordinate system. The dimension with the greatest variance is the first coordinate and is called the first principle component, the dimension with the second greatest variance is the second coordinate and is called the second principle component, an so on. PCA can also be used to reduce the number of dimensions by ignoring the dimensions with less variance. Performing PCA on the data could improve the results.

4. Dimension of Game Context

Using PCA could also provide insight that can be used to choose other dimensions that can be used. For example, the Boolean used in this thesis tracks whether or not there is a face card on the board. A Boolean for tracking the presence of an Ace and another that tracks the presence of a King could prove to be more useful. Also, a different technique for analyzing the board cards could be used. The board strength, probability of straight, probability of flush, probability of straight draw and probability of a flush draw dimensions used in this thesis could oversimplify the threats that a board presents to players.

C. CONCLUSIONS

Modeling modern adversaries is difficult because of the many, differing complexities on small terrorist groups. In order to be effective, one common system for modeling every group is necessary. This thesis attempts to create an adversary modeling system that is useful in the domain of Texas Hold’em Poker because of its structure, rules, and parallel with wartime adversarial situations. The results show that although the accuracy is not sufficient to return to the more complex domain of warfare, the Hidden Markov Models do perform significantly better than random guessing. With more modifications, the accuracy should improve enough to conduct experiments with terrorist models.

APPENDIX: RESULTS OF HMM EXPERIMENTS

A. EXPERIMENTS WITH FOUR HMMS

The first table applies to all of the other tables in Section A. It shows the number of predictions made for each group of actions.

Category	Number of Predictions
All Actions	1880
First Action	500
3rd Action	284
5th Action	113
6th Action	53
7th Action	18
8th Action	6
Last Action	500

Table 17. Number of Predictions in each Action Category.

All Actions					First Action				
Accuracy:	0.3851				Accuracy:	0.1900			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.2787	0.2544	0.2280	0.6754	Recall	0.0000	0.0000	0.2500	0.8077
Precision	0.8705	0.1160	0.1467	0.3627	Precision	0.0000	0.0000	0.1392	0.1995
F-score	0.4222	0.1593	0.1785	0.4720	F-score	0.0000	0.0000	0.1789	0.3200
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-41%	+39%	-4%	+6%	±Baseline	-100%	-100%	+11%	-7%
3rd Action					5th Action				
Accuracy:	0.3768				Accuracy:	0.5310			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1818	0.2963	0.1395	0.7019	Recall	0.5625	0.2500	0.2857	0.5873
Precision	0.5882	0.1194	0.1225	0.5448	Precision	0.9000	0.0417	0.1667	0.8222
F-score	0.2778	0.1702	0.1304	0.6135	F-score	0.6923	0.0714	0.2105	0.6852
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-50%	-2%	-50%	+14%	±Baseline	+57%	+4%	-5%	-4%
6th Action					7th Action				
Accuracy:	0.5660				Accuracy:	0.4444			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.6667	0.5000	0.4000	0.5588	Recall	0.7500	0.0000	0.0000	0.3571
Precision	0.8000	0.0909	0.1667	0.9500	Precision	1.0000	0.0000	0.0000	1.0000
F-score	0.7273	0.1539	0.2353	0.7037	F-score	0.8571	0.0000	0.0000	0.5263
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+97%	+112%	+36%	-10%	±Baseline	+136%	-100%	-100%	-40%
8th Action					Last Action				
Accuracy:	0.6667				Accuracy:	0.6080			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.6000	Recall	0.6462	0.3704	0.3182	0.6731
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.9767	0.2703	0.2258	0.3763
F-score	1.0000	0.0000	0.0000	0.7500	F-score	0.7778	0.3125	0.2642	0.4828
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444

Table 18. Results for the 50-centroid, 4-state HMMs.

All Actions				
Accuracy:	0.3596			
	Fold	Low	Med	High
Recall	0.2604	0.5790	0.2902	0.5299
Precision	0.9091	0.1038	0.1518	0.4914
F-score	0.4048	0.1760	0.1993	0.5099
Baseline	0.7110	0.1143	0.1862	0.4437
±Baseline	-43%	+54%	+7%	+15%

First Action				
Accuracy:	0.1720			
	Fold	Low	Med	High
Recall	0.0831	0.7407	0.2500	0.2692
Precision	0.6750	0.0654	0.1392	0.3733
F-score	0.1480	0.1201	0.1789	0.3129
Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-81%	+17%	+11%	-9%

3rd Action				
Accuracy:	0.3556			
	Fold	Low	Med	High
Recall	0.0546	0.6296	0.3256	0.6154
Precision	0.8571	0.1809	0.1867	0.5926
F-score	0.1026	0.2810	0.2373	0.6038
Baseline	0.5584	0.1736	0.2630	0.5361
±Baseline	-82%	+62%	-10%	+13%

5th Action				
Accuracy:	0.4956			
	Fold	Low	Med	High
Recall	0.5000	0.0000	0.3571	0.5556
Precision	1.0000	0.0000	0.1429	0.7778
F-score	0.6667	0.0000	0.2041	0.6482
Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	+51%	-100%	-7%	-9%

6th Action				
Accuracy:	0.5660			
	Fold	Low	Med	High
Recall	0.5833	1.0000	0.6000	0.5294
Precision	1.0000	0.2857	0.1667	0.8571
F-score	0.7368	0.4444	0.2609	0.6546
Baseline	0.3692	0.0727	0.1724	0.7816
±Baseline	+100%	+511%	+51%	-16%

7th Action				
Accuracy:	0.3333			
	Fold	Low	Med	High
Recall	0.7500	0.0000	0.0000	0.2143
Precision	1.0000	0.0000	0.0000	1.0000
F-score	0.8571	0.0000	0.0000	0.3529
Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+136%	-100%	-100%	-60%

8th Action				
Accuracy:	0.6667			
	Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.6000
Precision	1.0000	0.0000	0.0000	1.0000
F-score	1.0000	0.0000	0.0000	0.7500
Baseline	0.2857	2.0000	2.0000	0.9091
±Baseline	+250%	-100%	-100%	-17%

Last Action				
Accuracy:	0.5980			
	Fold	Low	Med	High
Recall	0.6462	0.4444	0.3636	0.5865
Precision	0.9906	0.3243	0.1861	0.3697
F-score	0.7821	0.3750	0.2462	0.4535
Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-1%	+266%	+52%	+32%

Table 19. Results for the 50-centroid, 8-state HMMs.

All Actions					First Action				
Accuracy:	0.4734				Accuracy:	0.5300			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.5217	0.3421	0.3731	0.4440	Recall	0.7477	0.0000	0.4318	0.0289
Precision	0.7807	0.1423	0.1548	0.5313	Precision	0.7023	0.0000	0.1310	0.3333
F-score	0.6254	0.2010	0.2188	0.4837	F-score	0.7243	0.0000	0.2011	0.0531
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-12%	+76%	+18%	+9%	±Baseline	-8%	-100%	+24%	-85%
3rd Action					5th Action				
Accuracy:	0.4014				Accuracy:	0.4779			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.2000	0.4815	0.3256	0.6250	Recall	0.5313	0.0000	0.2857	0.5238
Precision	0.6875	0.2000	0.1892	0.5752	Precision	0.7727	0.0000	0.1539	0.7674
F-score	0.3099	0.2826	0.2393	0.5991	F-score	0.6296	0.0000	0.2000	0.6226
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-45%	+63%	-9%	+12%	±Baseline	+43%	-100%	-9%	-13%
6th Action					7th Action				
Accuracy:	0.5472				Accuracy:	0.3333			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.5833	0.0000	0.6000	0.5588	Recall	0.7500	0.0000	0.0000	0.2143
Precision	0.7778	0.0000	0.1875	0.8636	Precision	0.7500	0.0000	0.0000	1.0000
F-score	0.6667	0.0000	0.2857	0.6786	F-score	0.7500	0.0000	0.0000	0.3529
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+81%	-100%	+66%	-13%	±Baseline	+106%	-100%	-100%	-60%
8th Action					Last Action				
Accuracy:	0.3333				Accuracy:	0.6120			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.2000	Recall	0.6800	0.3704	0.3864	0.5577
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.9526	0.2941	0.2024	0.3867
F-score	1.0000	0.0000	0.0000	0.3333	F-score	0.7935	0.3279	0.2656	0.4567
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-63%	±Baseline	+1%	+220%	+64%	+33%

Table 20. Results for 75-centroid, 4-state HMMs.

All Actions					First Action				
Accuracy:	0.3957				Accuracy:	0.2720			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3337	0.4912	0.3109	0.5261	Recall	0.2585	0.5926	0.2500	0.2404
Precision	0.8607	0.1024	0.1546	0.5193	Precision	0.7778	0.0672	0.1264	0.3731
F-score	0.4809	0.1694	0.2065	0.5227	F-score	0.3880	0.1208	0.1679	0.2924
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-32%	+48%	+11%	+18%	±Baseline	-51%	+18%	+4%	-15%
3rd Action					5th Action				
Accuracy:	0.3838				Accuracy:	0.5044			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1455	0.5185	0.3023	0.6346	Recall	0.5000	0.0000	0.2857	0.5873
Precision	0.6957	0.1944	0.1806	0.5641	Precision	0.7619	0.0000	0.1667	0.7708
F-score	0.2406	0.2828	0.2261	0.5973	F-score	0.6038	0.0000	0.2105	0.6667
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-57%	+63%	-14%	+11%	±Baseline	+37%	-100%	-5%	-7%
6th Action					7th Action				
Accuracy:	0.5660				Accuracy:	0.5000			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.5833	0.0000	0.6000	0.5882	Recall	0.7500	0.0000	0.0000	0.4286
Precision	0.7778	0.0000	0.2308	0.8333	Precision	0.7500	0.0000	0.0000	1.0000
F-score	0.6667	0.0000	0.3333	0.6897	F-score	0.7500	0.0000	0.0000	0.6000
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+81%	-100%	+93%	-12%	±Baseline	+106%	-100%	-100%	-31%
8th Action					Last Action				
Accuracy:	0.5000				Accuracy:	0.5980			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.4000	Recall	0.6523	0.3704	0.3409	0.5962
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.9507	0.2564	0.1807	0.4000
F-score	1.0000	0.0000	0.0000	0.5714	F-score	0.7737	0.3030	0.2362	0.4788
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-37%	±Baseline	-2%	+196%	+46%	+39%

Table 21. Results for 75-centroid, 8-state HMMs.

All Actions					First Action				
Accuracy:	0.4165				Accuracy:	0.3200			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3992	0.3860	0.3782	0.4702	Recall	0.3785	0.3333	0.4318	0.0865
Precision	0.7992	0.1007	0.1584	0.5431	Precision	0.7455	0.0529	0.1387	0.3214
F-score	0.5325	0.1597	0.2232	0.5040	F-score	0.5020	0.0914	0.2099	0.1364
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-25%	+40%	+20%	+14%	±Baseline	-36%	-11%	+30%	-60%
3rd Action					5th Action				
Accuracy:	0.3556				Accuracy:	0.5487			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1273	0.4444	0.3256	0.5865	Recall	0.5625	0.2500	0.4286	0.5873
Precision	0.5185	0.1600	0.1842	0.5755	Precision	0.7500	0.0769	0.1936	0.8222
F-score	0.2044	0.2353	0.2353	0.5810	F-score	0.6429	0.1177	0.2667	0.6852
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-63%	+36%	-11%	+8%	±Baseline	+46%	+72%	+21%	-4%
6th Action					7th Action				
Accuracy:	0.6226				Accuracy:	0.5000			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.6667	0.0000	0.6000	0.6471	Recall	0.7500	0.0000	0.0000	0.4286
Precision	0.8000	0.0000	0.2143	0.8800	Precision	0.7500	0.0000	0.0000	1.0000
F-score	0.7273	0.0000	0.3158	0.7458	F-score	0.7500	0.0000	0.0000	0.6000
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+97%	-100%	+83%	-5%	±Baseline	+106%	-100%	-100%	-31%
8th Action					Last Action				
Accuracy:	0.5000				Accuracy:	0.6240			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.4000	Recall	0.7015	0.2963	0.2727	0.6154
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.9268	0.2759	0.1622	0.4238
F-score	1.0000	0.0000	0.0000	0.5714	F-score	0.7986	0.2857	0.2034	0.5020
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-37%	±Baseline	+1%	+179%	+26%	+46%

Table 22. Results for 100-centroid, 4-state HMMs.

All Actions					First Action				
Accuracy:	0.5122				Accuracy:	0.5520			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.5641	0.2105	0.3368	0.5392	Recall	0.7231	0.0000	0.2500	0.2885
Precision	0.7657	0.1491	0.1625	0.5207	Precision	0.7015	0.0000	0.1358	0.3571
F-score	0.6496	0.1746	0.2192	0.5298	F-score	0.7121	0.0000	0.1760	0.3192
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-9%	+53%	+18%	+19%	±Baseline	-10%	-100%	+9%	-7%
3rd Action					5th Action				
Accuracy:	0.3908				Accuracy:	0.5133			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1455	0.4074	0.3721	0.6539	Recall	0.5313	0.0000	0.3571	0.5714
Precision	0.5333	0.2200	0.1861	0.5763	Precision	0.7083	0.0000	0.1667	0.7660
F-score	0.2286	0.2857	0.2481	0.6126	F-score	0.6071	0.0000	0.2273	0.6546
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-59%	+65%	-6%	+14%	±Baseline	+38%	-100%	+3%	-9%
6th Action					7th Action				
Accuracy:	0.6038				Accuracy:	0.5556			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.6667	0.0000	0.4000	0.6471	Recall	0.7500	0.0000	0.0000	0.5000
Precision	0.8889	0.0000	0.1333	0.8462	Precision	0.7500	0.0000	0.0000	1.0000
F-score	0.7619	0.0000	0.2000	0.7333	F-score	0.7500	0.0000	0.0000	0.6667
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+106%	-100%	+16%	-6%	±Baseline	+106%	-100%	-100%	-24%
8th Action					Last Action				
Accuracy:	0.3333				Accuracy:	0.6400			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.2000	Recall	0.7108	0.3333	0.3636	0.6154
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.9352	0.3913	0.2000	0.4267
F-score	1.0000	0.0000	0.0000	0.3333	F-score	0.8077	0.3600	0.2581	0.5039
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-63%	±Baseline	+3%	+251%	+60%	+46%

Table 23. Results for 100-centroid, 8-state HMMs.

All Actions					First Action				
Accuracy:	0.3803				Accuracy:	0.2480			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3317	0.3597	0.4197	0.4646	Recall	0.2492	0.5556	0.4318	0.0865
Precision	0.8191	0.0905	0.1543	0.5166	Precision	0.7714	0.0652	0.1387	0.3214
F-score	0.4722	0.1446	0.2256	0.4892	F-score	0.3767	0.1167	0.2099	0.1364
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-34%	+26%	+21%	+10%	±Baseline	-52%	+14%	+30%	-60%
3rd Action					5th Action				
Accuracy:	0.3873				Accuracy:	0.4690			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1273	0.3704	0.4186	0.6539	Recall	0.6250	0.0000	0.2143	0.4762
Precision	0.7368	0.1639	0.1978	0.6018	Precision	0.6061	0.0000	0.1154	0.7143
F-score	0.2171	0.2273	0.2687	0.6267	F-score	0.6154	0.0000	0.1500	0.5714
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-61%	+31%	+2%	+17%	±Baseline	+39%	-100%	-32%	-20%
6th Action					7th Action				
Accuracy:	0.5660				Accuracy:	0.3333			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.8333	0.0000	0.6000	0.5000	Recall	0.7500	0.0000	0.0000	0.2143
Precision	0.6250	0.0000	0.2727	0.8947	Precision	0.6000	0.0000	0.0000	1.0000
F-score	0.7143	0.0000	0.3750	0.6415	F-score	0.6667	0.0000	0.0000	0.3529
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+93%	-100%	+118%	-18%	±Baseline	+83%	-100%	-100%	-60%
8th Action					Last Action				
Accuracy:	0.5000				Accuracy:	0.5880			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.4000	Recall	0.6585	0.2963	0.3636	0.5385
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.8735	0.2759	0.2025	0.3810
F-score	1.0000	0.0000	0.0000	0.5714	F-score	0.7509	0.2857	0.2602	0.4462
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-37%	±Baseline	-5%	+179%	+61%	+30%

Table 24. Results for 175-centroid, 4-state HMMs.

All Actions					First Action				
Accuracy:	0.4032				Accuracy:	0.2660			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3539	0.4474	0.3627	0.5037	Recall	0.2492	0.5556	0.3409	0.2115
Precision	0.7996	0.1014	0.1795	0.5114	Precision	0.7714	0.0652	0.1515	0.3333
F-score	0.4906	0.1653	0.2401	0.5075	F-score	0.3767	0.1167	0.2098	0.2588
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-31%	+45%	+29%	+14%	±Baseline	-52%	+14%	+30%	-25%
3rd Action					5th Action				
Accuracy:	0.3908				Accuracy:	0.5044			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1455	0.4815	0.3954	0.6250	Recall	0.6250	0.0000	0.2143	0.5397
Precision	0.6400	0.1781	0.2180	0.6019	Precision	0.5405	0.0000	0.1364	0.7391
F-score	0.2370	0.2600	0.2810	0.6132	F-score	0.5797	0.0000	0.1667	0.6239
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-58%	+50%	+7%	+14%	±Baseline	+31%	-100%	-24%	-13%
6th Action					7th Action				
Accuracy:	0.6038				Accuracy:	0.4444			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.8333	0.0000	0.4000	0.5882	Recall	0.7500	0.0000	0.0000	0.3571
Precision	0.6250	0.0000	0.2222	0.8696	Precision	0.6000	0.0000	0.0000	1.0000
F-score	0.7143	0.0000	0.2857	0.7018	F-score	0.6667	0.0000	0.0000	0.5263
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+93%	-100%	+66%	-10%	±Baseline	+83%	-100%	-100%	-40%
8th Action					Last Action				
Accuracy:	0.5000				Accuracy:	0.5860			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.4000	Recall	0.6708	0.2593	0.2955	0.5289
Precision	0.5000	0.0000	0.0000	1.0000	Precision	0.8516	0.2333	0.1806	0.3873
F-score	0.6667	0.0000	0.0000	0.5714	F-score	0.7504	0.2456	0.2241	0.4472
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+133%	-100%	-100%	-37%	±Baseline	-5%	+140%	+39%	+30%

Table 25. Results for 175-centroid, 8-state HMMs.

All Actions					First Action				
Accuracy:	0.4112				Accuracy:	0.3120			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3828	0.4825	0.3057	0.4888	Recall	0.3292	0.5185	0.3409	0.1923
Precision	0.7505	0.1148	0.1761	0.4879	Precision	0.7040	0.0745	0.1515	0.3279
F-score	0.5070	0.1855	0.2235	0.4884	F-score	0.4486	0.1302	0.2098	0.2424
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-29%	+62%	+20%	+10%	±Baseline	-43%	+27%	+30%	-30%
3rd Action					5th Action				
Accuracy:	0.3944				Accuracy:	0.5044			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.1273	0.5556	0.2326	0.7019	Recall	0.6875	0.0000	0.3571	0.4762
Precision	0.4375	0.1807	0.2326	0.5794	Precision	0.5500	0.0000	0.1923	0.8108
F-score	0.1972	0.2727	0.2326	0.6348	F-score	0.6111	0.0000	0.2500	0.6000
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-65%	+57%	-12%	+18%	±Baseline	+38%	-100%	+13%	-16%
6th Action					7th Action				
Accuracy:	0.4906				Accuracy:	0.1667			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.8333	0.0000	0.4000	0.4118	Recall	0.7500	0.0000	0.0000	0.0000
Precision	0.5882	0.0000	0.1667	0.8235	Precision	0.6000	0.0000	0.0000	0.0000
F-score	0.6897	0.0000	0.2353	0.5490	F-score	0.6667	0.0000	0.0000	0.0000
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+87%	-100%	+36%	-30%	±Baseline	+83%	-100%	-100%	-100%
8th Action					Last Action				
Accuracy:	0.3333				Accuracy:	0.6000			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.2000	Recall	0.7015	0.3704	0.3182	0.4615
Precision	1.0000	0.0000	0.0000	1.0000	Precision	0.8291	0.4546	0.2090	0.3529
F-score	1.0000	0.0000	0.0000	0.3333	F-score	0.7600	0.4082	0.2523	0.4000
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+250%	-100%	-100%	-63%	±Baseline	-4%	+298%	+56%	+16%

Table 26. Results for 250-centroid, 4-state HMMs.

All Actions					First Action				
Accuracy:	0.3745				Accuracy:	0.2320			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.3713	0.4474	0.2228	0.4198	Recall	0.1908	0.7037	0.3182	0.2019
Precision	0.6696	0.0996	0.1311	0.4839	Precision	0.6739	0.0754	0.1489	0.3387
F-score	0.4777	0.1629	0.1651	0.4496	F-score	0.2974	0.1362	0.2029	0.2530
Baseline	0.7110	0.1143	0.1862	0.4437	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	-33%	+43%	-11%	+1%	±Baseline	-62%	+33%	+25%	-27%
3rd Action					5th Action				
Accuracy:	0.3908				Accuracy:	0.4425			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.2364	0.4444	0.2558	0.5962	Recall	0.8125	0.0000	0.2143	0.3333
Precision	0.5306	0.1818	0.1549	0.6327	Precision	0.4407	0.0000	0.1765	0.7000
F-score	0.3270	0.2581	0.1930	0.6139	F-score	0.5714	0.0000	0.1936	0.4516
Baseline	0.5584	0.1736	0.2630	0.5361	Baseline	0.4414	0.0684	0.2205	0.7159
±Baseline	-41%	+49%	-27%	+15%	±Baseline	+29%	-100%	-12%	-37%
6th Action					7th Action				
Accuracy:	0.4906				Accuracy:	0.4444			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	0.8333	0.0000	0.2000	0.4412	Recall	1.0000	0.0000	0.0000	0.2857
Precision	0.4348	0.0000	0.2500	0.7895	Precision	0.4444	0.0000	0.0000	1.0000
F-score	0.5714	0.0000	0.2222	0.5660	F-score	0.6154	0.0000	0.0000	0.4444
Baseline	0.3692	0.0727	0.1724	0.7816	Baseline	0.3636	2.0000	2.0000	0.8750
±Baseline	+55%	-100%	+29%	-28%	±Baseline	+69%	-100%	-100%	-49%
8th Action					Last Action				
Accuracy:	0.3333				Accuracy:	0.5740			
	Fold	Low	Med	High		Fold	Low	Med	High
Recall	1.0000	0.0000	0.0000	0.2000	Recall	0.7385	0.1852	0.1364	0.3462
Precision	0.2000	0.0000	0.0000	1.0000	Precision	0.7524	0.1613	0.1304	0.3462
F-score	0.3333	0.0000	0.0000	0.3333	F-score	0.7453	0.1724	0.1333	0.3462
Baseline	0.2857	2.0000	2.0000	0.9091	Baseline	0.7879	0.1025	0.1618	0.3444
±Baseline	+17%	-100%	-100%	-63%	±Baseline	-5%	+68%	-18%	+1%

Table 27. Results for 500-centroid, 8 state HMMs.

B. EXPERIMENTS WITH THREE HMMS

The first table applies to all of the other tables in Section B. It shows the number of predictions made for each group of actions.

Category	Number of Predictions
All Actions	843
First Action	175
3rd Action	174
5th Action	81
6th Action	41
7th Action	14
8th Action	5
Last Action	175

Table 28. Number of Predictions in each Action Category.

All Actions				First Action			
Accuracy:	0.5302			Accuracy:	0.5429		
	Low	Med	High		Low	Med	High
Recall	0.2544	0.2383	0.6940	Recall	0.0000	0.2500	0.8077
Precision	0.2197	0.2690	0.6889	Precision	0.0000	0.3333	0.5916
F-score	0.2358	0.2528	0.6915	F-score	0.0000	0.2857	0.6829
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	-1%	-32%	-11%	±Baseline	-100%	-29%	-8%
3rd Action				5th Action			
Accuracy:	0.5402			Accuracy:	0.5185		
	Low	Med	High		Low	Med	High
Recall	0.2963	0.1628	0.7596	Recall	0.2500	0.2857	0.5873
Precision	0.2581	0.2188	0.7117	Precision	0.0500	0.2353	0.8409
F-score	0.2759	0.1867	0.7349	F-score	0.0833	0.2581	0.6916
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+3%	-53%	-2%	±Baseline	-11%	-12%	-21%
6th Action				7th Action			
Accuracy:	0.5366			Accuracy:	0.3571		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.4000	0.5588	Recall	0.0000	0.0000	0.3571
Precision	0.1000	0.2000	0.9048	Precision	0.0000	0.0000	1.0000
F-score	0.1667	0.2667	0.6909	F-score	0.0000	0.0000	0.5263
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+79%	+23%	-24%	±Baseline	-100%	-100%	-47%
8th Action				Last Action			
Accuracy:	0.6000			Accuracy:	0.5371		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.6000	Recall	0.3704	0.3182	0.6731
Precision	0.0000	0.0000	1.0000	Precision	0.3030	0.3111	0.7217
F-score	0.0000	0.0000	0.7500	F-score	0.3333	0.3146	0.6965
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-25%	±Baseline	+25%	-22%	-7%

Table 29. Results for 50-Centroid, 4-state HMMs.

All Actions				First Action			
Accuracy:	0.4935			Accuracy:	0.3771		
	Low	Med	High		Low	Med	High
Recall	0.6053	0.2902	0.5429	Recall	0.7407	0.2500	0.3365
Precision	0.2961	0.2523	0.7500	Precision	0.2222	0.3333	0.6731
F-score	0.3977	0.2699	0.6299	F-score	0.3419	0.2857	0.4487
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+67%	-28%	-19%	±Baseline	+28%	-29%	-40%
3rd Action				5th Action			
Accuracy:	0.5460			Accuracy:	0.4938		
	Low	Med	High		Low	Med	High
Recall	0.6296	0.3256	0.6154	Recall	0.0000	0.3571	0.5556
Precision	0.4048	0.3044	0.7442	Precision	0.0000	0.1724	0.8537
F-score	0.4928	0.3146	0.6737	F-score	0.0000	0.2326	0.6731
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+83%	-21%	-10%	±Baseline	-100%	-21%	-23%
6th Action				7th Action			
Accuracy:	0.5610			Accuracy:	0.2143		
	Low	Med	High		Low	Med	High
Recall	1.0000	0.6000	0.5294	Recall	0.0000	0.0000	0.2143
Precision	0.4000	0.1765	0.9474	Precision	0.0000	0.0000	1.0000
F-score	0.5714	0.2727	0.6793	F-score	0.0000	0.0000	0.3529
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+514%	+25%	-25%	±Baseline	-100%	-100%	-65%
8th Action				Last Action			
Accuracy:	0.6000			Accuracy:	0.5086		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.6000	Recall	0.4444	0.3636	0.5865
Precision	0.0000	0.0000	1.0000	Precision	0.3750	0.2581	0.7531
F-score	0.0000	0.0000	0.7500	F-score	0.4068	0.3019	0.6595
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-25%	±Baseline	+52%	-25%	-12%

Table 30. Results for 50-centroid, 8-state HMMs.

All Actions				First Action			
Accuracy:	0.4567			Accuracy:	0.2629		
	Low	Med	High		Low	Med	High
Recall	0.6053	0.3834	0.4515	Recall	0.8889	0.4318	0.0289
Precision	0.2760	0.2731	0.7516	Precision	0.2330	0.2879	0.5000
F-score	0.3791	0.3190	0.5641	F-score	0.3692	0.3455	0.0546
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+59%	-14%	-27%	±Baseline	+38%	-14%	-93%
3rd Action				5th Action			
Accuracy:	0.5460			Accuracy:	0.4691		
	Low	Med	High		Low	Med	High
Recall	0.5185	0.3488	0.6346	Recall	0.2500	0.2857	0.5238
Precision	0.5185	0.2830	0.7021	Precision	0.0500	0.1818	0.8462
F-score	0.5185	0.3125	0.6667	F-score	0.0833	0.2222	0.6471
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+93%	-21%	-11%	±Baseline	-11%	-25%	-26%
6th Action				7th Action			
Accuracy:	0.5610			Accuracy:	0.2143		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.6000	0.5588	Recall	0.0000	0.0000	0.2143
Precision	0.1429	0.2308	0.9048	Precision	0.0000	0.0000	1.0000
F-score	0.2222	0.3333	0.6909	F-score	0.0000	0.0000	0.3529
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+139%	+53%	-24%	±Baseline	-100%	-100%	-65%
8th Action				Last Action			
Accuracy:	0.2000			Accuracy:	0.4971		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.2000	Recall	0.4444	0.3864	0.5577
Precision	0.0000	0.0000	1.0000	Precision	0.2857	0.3091	0.7436
F-score	0.0000	0.0000	0.3333	F-score	0.3478	0.3434	0.6374
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-67%	±Baseline	+30%	-15%	-15%

Table 31. Results for the 75-centroid, 4-state HMMs.

All Actions				First Action			
Accuracy:	0.4935			Accuracy:	0.3429		
	Low	Med	High		Low	Med	High
Recall	0.6228	0.3109	0.5317	Recall	0.8889	0.2500	0.2404
Precision	0.2806	0.2804	0.7580	Precision	0.2330	0.3056	0.6944
F-score	0.3869	0.2948	0.6250	F-score	0.3692	0.2750	0.3571
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	62%	-21%	-20%	±Baseline	+38%	-32%	-52%
3rd Action				5th Action			
Accuracy:	0.5575			Accuracy:	0.5185		
	Low	Med	High		Low	Med	High
Recall	0.5556	0.3023	0.6635	Recall	0.2500	0.2857	0.5873
Precision	0.5000	0.2766	0.7113	Precision	0.0556	0.2000	0.8605
F-score	0.5263	0.2889	0.6866	F-score	0.0909	0.2353	0.6981
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+96%	-27%	-8%	±Baseline	-3%	-20%	-20%
6th Action				7th Action			
Accuracy:	0.5854			Accuracy:	0.4286		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.6000	0.5882	Recall	0.0000	0.0000	0.4286
Precision	0.1250	0.2727	0.9091	Precision	0.0000	0.0000	1.0000
F-score	0.2000	0.3750	0.7143	F-score	0.0000	0.0000	0.6000
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+115%	+73%	-21%	±Baseline	-100%	-100%	-40%
8th Action				Last Action			
Accuracy:	0.4000			Accuracy:	0.5086		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.4000	Recall	0.4444	0.3409	0.5962
Precision	0.0000	0.0000	1.0000	Precision	0.3000	0.2885	0.7470
F-score	0.0000	0.0000	0.5714	F-score	0.3582	0.3125	0.6631
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-43%	±Baseline	+34%	-22%	-11%

Table 32. Results for the 75-centroid, 8-state HMMs.

All Actions				First Action			
Accuracy:	0.4781			Accuracy:	0.2971		
	Low	Med	High		Low	Med	High
Recall	0.5702	0.3834	0.4925	Recall	0.8889	0.4318	0.0865
Precision	0.2766	0.2835	0.7608	Precision	0.2400	0.3065	0.6923
F-score	0.3725	0.3260	0.5980	F-score	0.3780	0.3585	0.1539
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	56%	-13%	-23%	±Baseline	+41%	-11%	-79%
3rd Action				5th Action			
Accuracy:	0.5287			Accuracy:	0.5432		
	Low	Med	High		Low	Med	High
Recall	0.5185	0.3256	0.6154	Recall	0.2500	0.4286	0.5873
Precision	0.4000	0.2800	0.7191	Precision	0.0769	0.2308	0.8810
F-score	0.4516	0.3011	0.6632	F-score	0.1177	0.3000	0.7048
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+68%	-24%	-11%	±Baseline	+25%	+2%	-19%
6th Action				7th Action			
Accuracy:	0.6098			Accuracy:	0.4286		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.6000	0.6471	Recall	0.0000	0.0000	0.4286
Precision	0.0000	0.2500	0.8800	Precision	0.0000	0.0000	1.0000
F-score	0.0000	0.3529	0.7458	F-score	0.0000	0.0000	0.6000
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	-100%	+62%	-18%	±Baseline	-100%	-100%	-40%
8th Action				Last Action			
Accuracy:	0.4000			Accuracy:	0.4857		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.4000	Recall	0.2963	0.2955	0.6154
Precision	0.0000	0.0000	1.0000	Precision	0.2286	0.2600	0.7111
F-score	0.0000	0.0000	0.5714	F-score	0.2581	0.2766	0.6598
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-43%	±Baseline	-3%	-31%	-11%

Table 33. Results for 100-centroid, 4-state HMMs.

All Actions				First Action			
Accuracy:	0.5255			Accuracy:	0.4743		
	Low	Med	High		Low	Med	High
Recall	0.2807	0.3834	0.6287	Recall	0.0000	0.3636	0.6442
Precision	0.2883	0.2731	0.7310	Precision	0.0000	0.2388	0.6204
F-score	0.2844	0.3190	0.6760	F-score	0.0000	0.2883	0.6321
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	19%	-14%	-13%	±Baseline	-100%	-28%	-15%
3rd Action				5th Action			
Accuracy:	0.5805			Accuracy:	0.5062		
	Low	Med	High		Low	Med	High
Recall	0.4444	0.3954	0.6923	Recall	0.0000	0.3571	0.5714
Precision	0.5217	0.3269	0.7273	Precision	0.0000	0.2000	0.8571
F-score	0.4800	0.3579	0.7094	F-score	0.0000	0.2564	0.6857
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+79%	-10%	-5%	±Baseline	-100%	-13%	-22%
6th Action				7th Action			
Accuracy:	0.5854			Accuracy:	0.5000		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.4000	0.6471	Recall	0.0000	0.0000	0.5000
Precision	0.0000	0.1667	0.8462	Precision	0.0000	0.0000	1.0000
F-score	0.0000	0.2353	0.7333	F-score	0.0000	0.0000	0.6667
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	-100%	+8%	-19%	±Baseline	-100%	-100%	-33%
8th Action				Last Action			
Accuracy:	0.2000			Accuracy:	0.5086		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.2000	Recall	0.3333	0.3636	0.6154
Precision	0.0000	0.0000	1.0000	Precision	0.2647	0.2909	0.7442
F-score	0.0000	0.0000	0.3333	F-score	0.2951	0.3232	0.6737
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-67%	±Baseline	+10%	-20%	-10%

Table 34. Results for the 100-centroid, 8-state HMMs.

All Actions				First Action			
Accuracy:	0.4638			Accuracy:	0.2971		
	Low	Med	High		Low	Med	High
Recall	0.5088	0.4301	0.4664	Recall	0.8889	0.4318	0.0865
Precision	0.2437	0.2923	0.7788	Precision	0.2400	0.3065	0.6923
F-score	0.3296	0.3480	0.5834	F-score	0.3780	0.3585	0.1539
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+38%	-7%	-25%	±Baseline	+41%	-11%	-79%
3rd Action				5th Action			
Accuracy:	0.5575			Accuracy:	0.4321		
	Low	Med	High		Low	Med	High
Recall	0.4074	0.4186	0.6539	Recall	0.2500	0.2857	0.4762
Precision	0.3667	0.3214	0.7727	Precision	0.0500	0.1739	0.7895
F-score	0.3860	0.3636	0.7083	F-score	0.0833	0.2162	0.5941
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+44%	-8%	-5%	±Baseline	-11%	-27%	-32%
6th Action				7th Action			
Accuracy:	0.5122			Accuracy:	0.2143		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.6000	0.5000	Recall	0.0000	0.0000	0.2143
Precision	0.0909	0.2727	0.8947	Precision	0.0000	0.0000	1.0000
F-score	0.1539	0.3750	0.6415	F-score	0.0000	0.0000	0.3529
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+65%	+73%	-29%	±Baseline	-100%	-100%	-65%
8th Action				Last Action			
Accuracy:	0.4000			Accuracy:	0.4800		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.4000	Recall	0.4444	0.3636	0.5385
Precision	0.0000	0.0000	1.0000	Precision	0.2400	0.3077	0.7671
F-score	0.0000	0.0000	0.5714	F-score	0.3117	0.3333	0.6328
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-43%	±Baseline	+17%	-17%	-15%

Table 35. Results for the 175-centroids, 4-state HMMs.

All Actions				First Action			
Accuracy:	0.4982			Accuracy:	0.3486		
	Low	Med	High		Low	Med	High
Recall	0.5702	0.3731	0.5280	Recall	0.7778	0.3409	0.2404
Precision	0.2686	0.3064	0.7732	Precision	0.2283	0.3333	0.6579
F-score	0.3652	0.3365	0.6275	F-score	0.3529	0.3371	0.3521
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+53%	-10%	-19%	±Baseline	+32%	-16%	-53%
3rd Action				5th Action			
Accuracy:	0.5632			Accuracy:	0.5062		
	Low	Med	High		Low	Med	High
Recall	0.5185	0.3954	0.6442	Recall	0.2500	0.2857	0.5714
Precision	0.4375	0.3148	0.7614	Precision	0.0588	0.2000	0.8182
F-score	0.4746	0.3505	0.6979	F-score	0.0952	0.2353	0.6729
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+77%	-12%	-7%	±Baseline	+1%	-20%	-23%
6th Action				7th Action			
Accuracy:	0.5854			Accuracy:	0.3571		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.4000	0.6177	Recall	0.0000	0.0000	0.3571
Precision	0.1111	0.2500	0.8750	Precision	0.0000	0.0000	1.0000
F-score	0.1818	0.3077	0.7241	F-score	0.0000	0.0000	0.5263
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+95%	+42%	-20%	±Baseline	-100%	-100%	-47%
8th Action				Last Action			
Accuracy:	0.4000			Accuracy:	0.4686		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.4000	Recall	0.4074	0.2955	0.5577
Precision	0.0000	0.0000	1.0000	Precision	0.2340	0.2549	0.7533
F-score	0.0000	0.0000	0.5714	F-score	0.2973	0.2737	0.6409
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-43%	±Baseline	+11%	-32%	-14%

Table 36. Results for the 175-centroid, 8-state HMMs.

All Actions				First Action			
Accuracy:	0.4804			Accuracy:	0.3371		
	Low	Med	High		Low	Med	High
Recall	0.6491	0.3368	0.4963	Recall	0.8889	0.3409	0.1923
Precision	0.2509	0.3283	0.7600	Precision	0.2353	0.3333	0.7143
F-score	0.3619	0.3325	0.6005	F-score	0.3721	0.3371	0.3030
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+52%	-11%	-23%	±Baseline	+39%	-16%	-59%
3rd Action				5th Action			
Accuracy:	0.5920			Accuracy:	0.4691		
	Low	Med	High		Low	Med	High
Recall	0.5926	0.2558	0.7308	Recall	0.5000	0.4286	0.4762
Precision	0.3721	0.3667	0.7525	Precision	0.0800	0.2857	0.8571
F-score	0.4571	0.3014	0.7415	F-score	0.1379	0.3429	0.6122
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+70%	-24%	-1%	±Baseline	+47%	+16%	-30%
6th Action				7th Action			
Accuracy:	0.4146			Accuracy:	0.0000		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.4000	0.4118	Recall	0.0000	0.0000	0.0000
Precision	0.0833	0.1667	0.8235	Precision	0.0000	0.0000	0.0000
F-score	0.1429	0.2353	0.5490	F-score	0.0000	0.0000	0.0000
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+54%	+8%	-39%	±Baseline	-100%	-100%	-100%
8th Action				Last Action			
Accuracy:	0.2000			Accuracy:	0.4457		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.2000	Recall	0.5556	0.3409	0.4615
Precision	0.0000	0.0000	1.0000	Precision	0.2419	0.3125	0.7385
F-score	0.0000	0.0000	0.3333	F-score	0.3371	0.3261	0.5681
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-67%	±Baseline	+26%	-19%	-24%

Table 37. Results for the 250-centroid, 8-state HMMs.

All Actions				First Action			
Accuracy:	0.4081			Accuracy:	0.3371		
	Low	Med	High		Low	Med	High
Recall	0.5439	0.2539	0.4347	Recall	0.7037	0.4318	0.2019
Precision	0.1962	0.2322	0.7373	Precision	0.2568	0.2639	0.7241
F-score	0.2884	0.2426	0.5470	F-score	0.3762	0.3276	0.3158
Baseline	0.2382	0.3726	0.7774	Baseline	0.2673	0.4018	0.7455
±Baseline	+21%	-35%	-30%	±Baseline	+41%	-18%	-58%
3rd Action				5th Action			
Accuracy:	0.5057			Accuracy:	0.3210		
	Low	Med	High		Low	Med	High
Recall	0.4815	0.2558	0.6154	Recall	0.5000	0.2143	0.3333
Precision	0.2766	0.2619	0.7529	Precision	0.0556	0.1875	0.7241
F-score	0.3514	0.2588	0.6773	F-score	0.1000	0.2000	0.4565
Baseline	0.2687	0.3963	0.7482	Baseline	0.0941	0.2947	0.8750
±Baseline	+31%	-35%	-9%	±Baseline	+6%	-32%	-48%
6th Action				7th Action			
Accuracy:	0.4146			Accuracy:	0.3571		
	Low	Med	High		Low	Med	High
Recall	0.5000	0.2000	0.4412	Recall	0.0000	0.0000	0.3571
Precision	0.0526	0.2500	0.8333	Precision	0.0000	0.0000	1.0000
F-score	0.0952	0.2222	0.5769	F-score	0.0000	0.0000	0.5263
Baseline	0.0930	0.2174	0.9067	Baseline	2.0000	2.0000	1.0000
±Baseline	+2%	+2%	-36%	±Baseline	-100%	-100%	-47%
8th Action				Last Action			
Accuracy:	0.6000			Accuracy:	0.3143		
	Low	Med	High		Low	Med	High
Recall	0.0000	0.0000	0.6000	Recall	0.4074	0.1364	0.3654
Precision	0.0000	0.0000	1.0000	Precision	0.1264	0.2143	0.6333
F-score	0.0000	0.0000	0.7500	F-score	0.1930	0.1667	0.4634
Baseline	2.0000	2.0000	1.0000	Baseline	0.2673	0.4018	0.7455
±Baseline	-100%	-100%	-25%	±Baseline	-28%	-59%	-38%

Table 38. Results for the 500-centroid, 8-state HMMs.

C. EXPERIMENTS WITH TWO HMMS

The first table applies to all of the other tables in Section C. It shows the number of predictions made for each group of actions. All HMMs in Section C contained eight states.

Category	Number of Predictions
All Actions	1880
First Action	500
3rd Action	284
Last Action	500

Table 39. Number of Predictions in each Action Category.

All Actions			First Action		
Accuracy	0.6718		Accuracy	0.6200	
	Negative	Positive		Negative	Positive
Recall	0.6868	0.6596	Recall	0.4286	0.7231
Precision	0.6212	0.7215	Precision	0.4546	0.7015
F-score	0.6524	0.6892	F-score	0.4412	0.7121
Baseline	0.6192	0.7110	Baseline	0.5185	0.7879
±Baseline	+5%	-3%	±Baseline	-15%	-10%
Third Action			Last Action		
Accuracy	0.5915		Accuracy	0.7800	
	Negative	Positive		Negative	Positive
Recall	0.7644	0.3182	Recall	0.8686	0.7323
Precision	0.6394	0.4605	Precision	0.6360	0.9119
F-score	0.6963	0.3763	F-score	0.7343	0.8123
Baseline	0.7598	0.5584	Baseline	0.5185	0.7879
±Baseline	-8%	-33%	±Baseline	+42%	+3%

Table 40. Results for 100-centroid HMMs predicting fold or not-fold.

All Actions			First Action		
Accuracy	0.6436		Accuracy	0.6620	
	Negative	Positive		Negative	Positive
Recall	0.6064	0.7369	Recall	0.7096	0.4808
Precision	0.8525	0.4275	Precision	0.8388	0.3030
F-score	0.7087	0.5411	F-score	0.7688	0.3718
Baseline	0.8338	0.4437	Baseline	0.8839	0.3444
±Baseline	-15%	+22%	±Baseline	-13%	+8%
Third Action			Last Action		
Accuracy	0.6197		Accuracy	0.6520	
	Negative	Positive		Negative	Positive
Recall	0.5056	0.8173	Recall	0.6111	0.8077
Precision	0.8273	0.4885	Precision	0.9237	0.3529
F-score	0.6276	0.6115	F-score	0.7356	0.4912
Baseline	0.7759	0.5361	Baseline	0.8839	0.3444
±Baseline	-19%	+14%	±Baseline	-16%	+43%

Table 41. Results for 100-centroid HMMs predicting high or not-high.

All Actions			First Action		
Accuracy	0.6117		Accuracy	0.7040	
	Negative	Positive		Negative	Positive
Recall	0.6159	0.5751	Recall	0.7259	0.4773
Precision	0.9269	0.1463	Precision	0.9350	0.1438
F-score	0.7400	0.2332	F-score	0.8173	0.2211
Baseline	0.9459	0.1862	Baseline	0.9540	0.1618
±Baseline	-22%	+25%	±Baseline	-14%	+37
Third Action			Last Action		
Accuracy	0.4894		Accuracy	0.6780	
	Negative	Positive		Negative	Positive
Recall	0.4730	0.5814	Recall	0.6864	0.5909
Precision	0.8636	0.1645	Precision	0.9456	0.1539
F-score	0.6113	0.2564	F-score	0.7954	0.2441
Baseline	0.9181	0.2630	Baseline	0.9540	0.1618
±Baseline	-33%	-2%	±Baseline	-17%	+51%

Table 42. Results for 100-centroid HMMs predicting medium or not-medium.

All Actions			First Action		
Accuracy	0.6032		Accuracy	0.3900	
	Negative	Positive		Negative	Positive
Recall	0.6053	0.5702	Recall	0.3679	0.7778
Precision	0.9562	0.0853	Precision	0.9667	0.0656
F-score	0.7413	0.1484	F-score	0.5329	0.1210
Baseline	0.9687	0.1143	Baseline	0.9723	0.1025
±Baseline	-23%	+30%	±Baseline	-45%	+18%
Third Action			Last Action		
Accuracy	0.5669		Accuracy	0.8460	
	Negative	Positive		Negative	Positive
Recall	0.5681	0.5556	Recall	0.8710	0.4074
Precision	0.9241	0.1191	Precision	0.9626	0.1528
F-score	0.7036	0.1961	F-score	0.9145	0.2222
Baseline	0.9501	0.1736	Baseline	0.9723	0.1025
±Baseline	-26%	+13%	±Baseline	-6%	+117%

Table 43. Results for 100-centroid HMMs predicting low or not-low.

All Actions			First Action		
Accuracy	0.6612		Accuracy	0.6220	
	Negative	Positive		Negative	Positive
Recall	0.6892	0.6384	Recall	0.4229	0.7292
Precision	0.6077	0.7165	Precision	0.4568	0.7012
F-score	0.6459	0.6752	F-score	0.4392	0.7149
Baseline	0.6192	0.7110	Baseline	0.5185	0.7879
±Baseline	+4%	-5%	±Baseline	-15%	-9%
Third Action			Last Action		
Accuracy	0.6021		Accuracy	0.7700	
	Negative	Positive		Negative	Positive
Recall	0.7356	0.3909	Recall	0.7771	0.7662
Precision	0.6564	0.4832	Precision	0.6415	0.8646
F-score	0.6938	0.4322	F-score	0.7028	0.8124
Baseline	0.7598	0.5584	Baseline	0.5185	0.7879
±Baseline	-9%	-23%	±Baseline	+36%	+3%

Table 44. Results for 250-centroid HMMs predicting fold or not-fold.

All Actions			First Action		
Accuracy	0.6314		Accuracy	0.6640	
	Negative	Positive		Negative	Positive
Recall	0.6191	0.6623	Recall	0.7147	0.4712
Precision	0.8213	0.4095	Precision	0.8373	0.3025
F-score	0.7060	0.5061	F-score	0.7711	0.3684
Baseline	0.8338	0.4437	Baseline	0.8839	0.3444
Base	-15%	+14%	Base	-13%	+7%
Third Action			Last Action		
Accuracy	0.5845		Accuracy	0.6260	
	Negative	Positive		Negative	Positive
Recall	0.4556	0.8077	Recall	0.6162	0.6635
Precision	0.8039	0.4615	Precision	0.8746	0.3122
F-score	0.5816	0.5874	F-score	0.7230	0.4246
Baseline	0.7759	0.5361	Baseline	0.8839	0.3444
Base	-25%	+10%	Base	-18%	+23%

Table 45. Results for 250-centroid HMMs predicting high or not-high.

All Actions			First Action		
Accuracy	0.6261		Accuracy	0.7040	
	Negative	Positive		Negative	Positive
Recall	0.6343	0.5544	Recall	0.7259	0.4773
Precision	0.9256	0.1478	Precision	0.9350	0.1438
F-score	0.7527	0.2334	F-score	0.8173	0.2211
Baseline	0.9459	0.1862	Baseline	0.9540	0.1618
±Baseline	-20%	+25%	±Baseline	-14%	+37
Third Action			Last Action		
Accuracy	0.5246		Accuracy	0.7220	
	Negative	Positive		Negative	Positive
Recall	0.5228	0.5349	Recall	0.7303	0.6364
Precision	0.8630	0.1667	Precision	0.9542	0.1854
F-score	0.6512	0.2541	F-score	0.8273	0.2872
Baseline	0.9181	0.2630	Baseline	0.9540	0.1618
±Baseline	-29%	-3%	±Baseline	-13%	+78%

Table 46. Results for 250-centroid HMMs predicting medium or not-medium.

All Actions			First Action		
Accuracy	0.6553		Accuracy	0.4980	
	Negative	Positive		Negative	Positive
Recall	0.6648	0.5088	Recall	0.4968	0.5185
Precision	0.9545	0.0892	Precision	0.9476	0.0556
F-score	0.7837	0.1518	F-score	0.6519	0.1004
Baseline	0.9687	0.1143	Baseline	0.9723	0.1025
±Baseline	-19%	+33%	±Baseline	-33%	-2%
Third Action			Last Action		
Accuracy	0.6338		Accuracy	0.8560	
	Negative	Positive		Negative	Positive
Recall	0.6459	0.5185	Recall	0.8774	0.4815
Precision	0.9274	0.1333	Precision	0.9674	0.1831
F-score	0.7615	0.2121	F-score	0.9202	0.2653
Baseline	0.9501	0.1736	Baseline	0.9723	0.1025
±Baseline	-20%	+22%	±Baseline	-5%	+159%

Table 47. Results for 250-centroid HMMs predicting low or not-low.

All Actions			First Action		
Accuracy	0.6128		Accuracy	0.5640	
	Negative	Positive		Negative	Positive
Recall	0.6536	0.5796	Recall	0.4800	0.6092
Precision	0.5583	0.6730	Precision	0.3981	0.6851
F-score	0.6022	0.6228	F-score	0.4352	0.6450
Baseline	0.6192	0.7110	Baseline	0.5185	0.7879
±Baseline	-3%	-12%	±Baseline	-16%	-18%
Third Action			Last Action		
Accuracy	0.6338		Accuracy	0.7080	
	Negative	Positive		Negative	Positive
Recall	0.7989	0.3727	Recall	0.6057	0.7631
Precision	0.6683	0.5395	Precision	0.5792	0.7823
F-score	0.7278	0.4409	F-score	0.5922	0.7726
Baseline	0.7598	0.5584	Baseline	0.5185	0.7879
±Baseline	-4%	-21%	±Baseline	+14%	-2%

Table 48. Results for 500-centroid HMMs predicting fold or not-fold.

All Actions			First Action		
Accuracy	0.6420		Accuracy	0.6940	
	Negative	Positive		Negative	Positive
Recall	0.6414	0.6437	Recall	0.7727	0.3942
Precision	0.8186	0.4172	Precision	0.8293	0.3130
F-score	0.7192	0.5062	F-score	0.8000	0.3489
Baseline	0.8338	0.4437	Baseline	0.8839	0.3444
±Baseline	-14%	+14%	±Baseline	-9%	+1%
Third Action			Last Action		
Accuracy	0.6197		Accuracy	0.6120	
	Negative	Positive		Negative	Positive
Recall	0.5278	0.7789	Recall	0.5884	0.7019
Precision	0.8051	0.4880	Precision	0.8826	0.3093
F-score	0.6376	0.6000	F-score	0.7061	0.4294
Baseline	0.7759	0.5361	Baseline	0.8839	0.3444
±Baseline	-18%	+12%	±Baseline	-20%	+25%

Table 49. Results for 500-centroid HMMs predicting high or not-high.

All Actions			First Action		
Accuracy	0.6372		Accuracy	0.6940	
	Negative	Positive		Negative	Positive
Recall	0.6574	0.4611	Recall	0.7171	0.4546
Precision	0.9143	0.1334	Precision	0.9316	0.1342
F-score	0.7648	0.2070	F-score	0.8104	0.2073
Baseline	0.9459	0.1862	Baseline	0.9540	0.1618
±Baseline	-19%	+11%	±Baseline	-15%	+28%
Third Action			Last Action		
Accuracy	0.5739		Accuracy	0.6740	
	Negative	Positive		Negative	Positive
Recall	0.5975	0.4419	Recall	0.6864	0.5455
Precision	0.8571	0.1638	Precision	0.9399	0.1437
F-score	0.7042	0.2390	F-score	0.7934	0.2275
Baseline	0.9181	0.2630	Baseline	0.9540	0.1618
±Baseline	-23%	-9%	±Baseline	-17%	+41%

Table 50. Results for 500-centroid HMMs predicting medium or not-medium.

All Actions			First Action		
Accuracy	0.6011		Accuracy	0.4220	
	Negative	Positive		Negative	Positive
Recall	0.6053	0.5351	Recall	0.4038	0.7407
Precision	0.9528	0.0805	Precision	0.9647	0.0662
F-score	0.7403	0.1399	F-score	0.5693	0.1216
Baseline	0.9687	0.1143	Baseline	0.9723	0.1025
±Baseline	-14%	+22%	±Baseline	-41%	+19%
Third Action			Last Action		
Accuracy	0.6549		Accuracy	0.7260	
	Negative	Positive		Negative	Positive
Recall	0.6770	0.4444	Recall	0.7421	0.4444
Precision	0.9206	0.1263	Precision	0.9590	0.0896
F-score	0.7803	0.1967	F-score	0.8367	0.1491
Baseline	0.9501	0.1736	Baseline	0.9723	0.1025
±Baseline	-18%	+13%	±Baseline	-14%	+45%

Table 51. Results for 500-centroid HMMs predicting low or not-low.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] S. Griffith, *Sun Tzu: The Art of War*. Oxford: Oxford University Press, 1963, p 84.

[2] D. Billings. "The first international Roshambo programming competition," *International Computer Games Association Journal*, vol 23, pp. 42-50, 2000.

[3] D. Billings. "Thoughts on Roshambo," *International Computer Games Association Journal*, vol 23, pp. 3-8, 2000.

[4] M. Richards and Eyal Amir. "Opponent Modeling in Scrabble," in *International Joint Conference on Artificial Intelligence*, 2007, pp. 1483-1487.

[5] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron, "Approximating game-theoretic optimal strategies for full-scale Poker," in *International Conference on Artificial Intelligence*, 2003, pp 661-668.

[6] Darse Billings, "Algorithms and assessment in computer poker", Ph.D. dissertation, University of Alberta, Edmonton, Alberta, Canada, 2006.

[7] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron, "The challenge of poker," *Artificial Intelligence*, vol. 134, pp. 201-240, January 2002.

[8] D. Billings, D. Papp, J. Schaeffer and D. Szafron, "Opponent modeling in poker," in *American Association of Artificial Intelligence National Conference*, 1998, pp. 493-499.

[9] A. Davidson, D. Billings, J. Schaeffer, and D. Szafron, "Improved opponent modeling in poker," in *International Conference on Artificial Intelligence*, 2000, pp. 1467-1473.

[10] D. Billings, A. Davidson, T. Schauenberg, N. Burch, M. Bowling, R. Holte, J. Schaeffer, and D. Szafron, "Game tree search with adaptation in stochastic imperfect information games," in *Computers and Games: 4th International Conference*, 2004, pp. 21-34.

[11] T. Schauenberg, "Opponent modeling and search in poker," M.S. Thesis, University of Alberta, Edmonton, Alberta, Canada, 2006.

[12] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and C. Rayney, "Bayes' bluff: Opponent modeling in poker," in *21st Conference on Uncertainty in Artificial Intelligence*, 2005, pp. 550-558.

[13] A. Gilpin and T. Sandholm, "A Texas Hold'em poker player based on automated abstraction and real-time equilibrium computation," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2006, pp. 1453–1454.

[14] A. Gilpin and T. Sandholm, "Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em Poker," presented at *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Honolulu, Hawaii 2007.

[15] A. Gilpin, T. Sandholm, T. Sorensen, "Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em Poker," presented at *National Conference on Artificial Intelligence (AAAI)*, Vancouver, British Colombia, Canada, 2007.

[16] M. Terry and B. Mihok, "A bayesian net inference tool for hidden state in Texas Hold'em Poker," Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[17] K. Korb, A. Nicholson, N. Jitnah, "Bayesian poker," in *Uncertainty in Artificial Intelligence*, pp. 343-350, 1999.

[18] D. Boulton, "Bayesian opponent modeling in poker," Honours thesis, Monash University, 2003, http://www.csse.monash.edu.au/hons/projects/2000/Darren.Boulton/, September 2007.

[19] J. Carlton, "Bayesian poker, "Honours thesis," Monash University, 2000, http://www.csse.monash.edu.au/hons/projects/2000/Jason.Carlton/, September 2007.

[20] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Second Edition*. Prentice Hall, 1995.

[21] University of Alberta's Computer Poker Research Group, University of Alberta, "Michael Maurer's IRC poker database," http://games.cs.ualberta.ca/poker/IRC/, September 2007.

[22] University of Alberta's Computer Poker Research Group, University of Alberta, "University of Alberta's hand evaluator," http://spaz.ca/poker/UofAHandEval.zip, September 2007.

[23] D. Sklansky and M. Malmuth, *Hold'erm Poker for Advanced Players*. Two Plus Two Publishing, 2004, pp 14-17.

[24] Department of Computer Science and Engineering, University of California, San Diego, "Fast k-means code for Matlab," http://www.cse.ucsd.edu/users/elkan/fastkmeans.html, September 2007.

[25] Department of Computer Science, University of British Columbia, "Hidden Markov Model (HMM) Toolbox for Matlab," http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html, September 2007.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Kevin Squire
   Naval Postgraduate School
   Monterey, California

4. Dr. Craig Martell
   Naval Postgraduate School
   Monterey, California

5. Capt. Paul Avellino
   United States Marine Corps
   Naval Postgraduate School
   Monterey, California